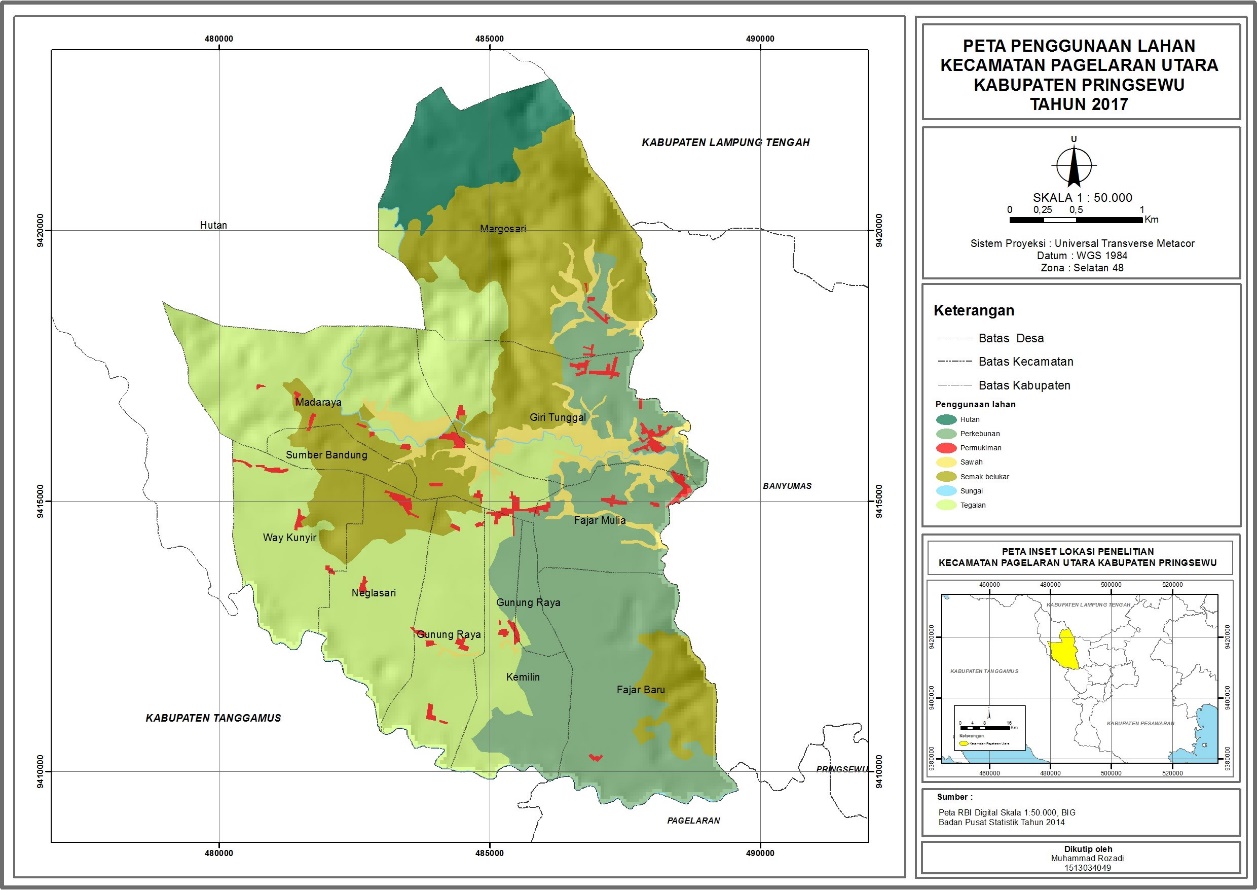
# **BAB II**

# **TINJAUAN LITERATUR**

# **Landasan Teori**

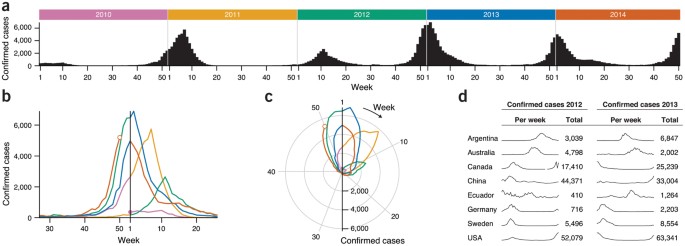
# **Data *Spatio-Temporal***

Data *spatio-temporal* terdiri dari representasi spasial dan temporal yang mengacu pada data yang dikumpulkan dan dianalisis melintasi kedua dimensi ruang dan waktu (Hamdi et al., 2022). Data spasial merepresentasikan objek spasial yang terdiri dari titik, garis, daerah, polygon sederhana, volume, bahkan data berdimensi tinggi yang mencakup waktu. Atribut data spasial mencakup kota, sungai, jalan raya, negara bagian, pegunungan, dan sebagainya. Contoh properti spasial mencakup luas sungai tertentu, batas wilayah, peta penggunaan lahan (Rozadi, 2019) yang ditunjukkan pada **Gambar 1**, atau informasi atribut non-spasial seperti ketinggian elevasi, nama kota, dan lainnya. Data spasial banyak digunakan dalam aplikasi pemantauan lingkungan, ruang, perencanaan kota, pengelolaan sumber daya, dan sistem informasi geografis (GIS) (Samet, 1995).



**Gambar 1.** Data Spasial: Peta Penggunaan Lahan Kecamatan Pagelaran Utara, Kabupaten Pringsewu Tahun 2017

Data temporal berkaitan dengan peristiwa yang diurutkan berdasarkan satu atau lebih dimensi waktu baik masa lalu, masa kini, maupun masa depan, seperti data Kasus Influenza yang Terkonfirmasi oleh WHO di Amerika Serikat Tahun 2010 – 2014 (World Health Organization, 2010) yang ditunjukkan pada **Gambar 2**. Data temporal dibedakan menjadi dua area secara luas. Yang pertama, berkaitan dengan hubungan sebab akibat di antara peristiwa-peristiwa yang berorientasi pada waktu. Yang kedua, berkaitan dengan penemuan pola serupa dalam rangkaian waktu yang sama atau di antara rangkaian waktu yang berbeda, disebut analisis deret waktu (*time series*) (Roddick & Spiliopoulou, 1999). Aspek penelitian analisis *time series* meliputi pendekatan kurva dengan metode matematika, pengurangan *noise*, perbandingan deret waktu menggunakan teknik pencocokan pola dan prediksi menggunakan metode matematika atau *neural network* (Weigend, 2018).



**Gambar 2.** Data Temporal: Kasus Influenza Terkonfirmasi di Amerika Serikat 2010 – 2014

Data *spatio-temporal* memiliki kompleksitas tinggi dan digunakan dalam berbagai bidang seperti meteorologi, ilmu lingkungan, kedokteran, transportasi, dan lainnya (Amran et al., 2020). Metode *data mining* dapat digunakan untuk mengolah dan menganalisis data *spatio-temporal* untuk berbagai tujuan, seperti visualisasi pemetaan sebaran penyakit, analisis geografis, pengukuran nilai tanah, hingga prediksi kualitas udara (Heldayani et al., 2021) (Asgari et al., 2022) (Alizanovic, 2023).

Prediksi merupakan salah satu metode dalam *Supervised Learning* untuk memperkirakan *output* di masa yang akan datang melalui proses pembelajaran atau pelatihan. Pembelajaran ini dilakukan berdasarkan data *input* dari masa lalu dan masa kiniyang telah diberikan dengan tujuan untuk meminimalkan kesalahan yang terjadi (Dewi et al., 2022). Data prediksi dapat berasal dari berbagai sumber, seperti data historis, data *spatio-temporal*, atau data yang terus-menerus diperbarui (Hartatik et al., 2023). Data ini kemudian digunakan untuk melatih model agar dapat mempelajari pola antara *input* dan *output*, sehingga dapat melakukan prediksi terhadap data baru.

# **Indeks Standar Pencemar Udara (ISPU)**

Indeks kualitas udara merupakan suatu parameter tanpa satuan yang memberikan informasi mengenai gambaran kondisi kualitas udara ambien di lokasi dan waktu tertentu kepada masyarakat. Indeks kualitas udara yang resmi digunakan di Indonesia berdasarkan Keputusan Menteri Negara Lingkungan Hidup Nomor 45 Tahun 1997 adalah Indeks Standar Pencemar Udara (ISPU) (Eko H., 2017). Penilaian ISPU didasarkan pada dampak yang dapat ditimbulkan terhadap kesehatan manusia, nilai estetika lingkungan, dan makhluk hidup lainnya. ISPU dapat dijadikan sebagai sistem peringatan dini (*early warning system*) bagi masyarakat yang tinggal di daerah rawan terdampak kebakaran hutan dan lahan. ISPU juga digunakan sebagai bahan pertimbangan untuk merumuskan upaya-upaya pengendalian pencemaran udara, baik oleh pemerintah pusat maupun daerah.

Terdapat tujuh parameter pencemar udara yang digunakan dalam perhitungan ISPU dan didasari pada besarnya risiko terhadap kesehatan manusia, yaitu PM10, PM2.5, NO2, SO2, CO, O3, dan HC (Chaniago et al., 2020). ISPU dihitung berdasarkan pada nilai-nilai ISPU batas atas dan bawah, konsentrasi ambien batas atas dan bawah, serta konsentrasi ambien dari hasil pengukuran. Persamaan perhitungan ISPU sebagai berikut:

(2.1)

Keterangan:

*I* = ISPU terhitung

*Ia*= ISPU batas atas

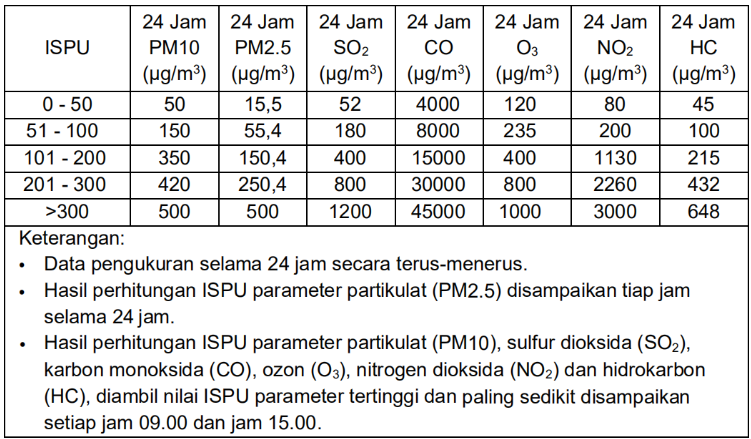
*Ib*= ISPU batas bawah

*Xa*= Konsentrasi ambien batas atas

*Xb*= Konsentrasi ambien batas bawah

*Xx*= Konsentrasi ambien nyata hasil pengukuran

Nilai batas atas ISPU paling tinggi adalah lebih dari 300, sedangkan nilai batas bawah ISPU paling rendah adalah 50. Kosentrasi ambien batas atas dan bawah berbeda pada tiap parameter dan diperoleh dari tabel konversi nilai konsentrasi parameter ISPU. Nilai konsentrasi ambien yang sebenarnya diperoleh dari rata-rata konsentrasi ambien selama 24 jam pengukuran (Firman et al., 2023). Tabel konversi nilai konsentrasi parameter ISPU dapat dilihat pada **Gambar 3** sebagai berikut:



**Gambar 3.** Tabel Konversi Nilai Konsentrasi Parameter ISPU

Berdasarkan Kementerian Lingkungan Hidup dan Kehutanan, ISPU terbagi menjadi lima kategori sesuai rentang nilainya (Wibawana, 2023), yaitu:

1. Kategori Baik (0 – 50)

ISPU dengan rentang nilai 0 – 50 memiliki kategori Baik. Tingkat kualitas udara masih sangat baik dan tidak memberikan efek negatif terhadap kesehatan manusia dan hewan, serta tidak berpengaruh terhadap tumbuhan, bangunan, dan nilai estetika.

1. Kategori Sedang (51 – 100)

ISPU dengan rentang nilai 51 – 100 memiliki kategori Sedang. Tingkat kualitas udara masih dapat diterima oleh kesehatan manusia dan hewan, tetapi sedikit berpengaruh terhadap tumbuhan yang sensitif dan nilai estetika.

1. Kategori Tidak Sehat (101 – 200)

ISPU dengan rentang nilai 101 – 200 memiliki kategori Tidak Sehat. Tingkat kualitas udara bersifat merugikan pada manusia maupun hewan atau dapat menimbulkan kerusakan pada tumbuhan dan nilai estetika.

1. Kategori Sangat Tidak Sehat (201 – 300)

ISPU dengan rentang nilai 201 – 300 memiliki kategori Sangat Tidak Sehat. Tingkat kualitas udara dapat menyebabkan kerugian dan meningkatkan risiko kesehatan pada beberapa kelompok populasi yang terpapar.

1. Kategori Berbahaya (300+)

ISPU dengan rentang nilai lebih dari 300 memiliki kategori Berbahaya. Tingkat kualitas udara berbahaya dan secara umum dapat menyebabkan kerugian kesehatan serius pada populasi dan memerlukan penanganan dengan cepat.

Data ISPU diperoleh dari pengoperasian Stasiun Pemantauan Kualitas Udara Ambien dan disampaikan kepada masyarakat setiap 24 jam dari data sebelumnya dan berlaku 24 jam ke depan. Waktu pengambilan data terakhir dilakukan pada pukul 15.00 WIB. ISPU dapat diakses melalui aplikasi ISPU Net yang memungkinkan masyarakat mengetahui kondisi kualitas udara secara *real-time* di seluruh wilayah (Ramadhan P, 2021).

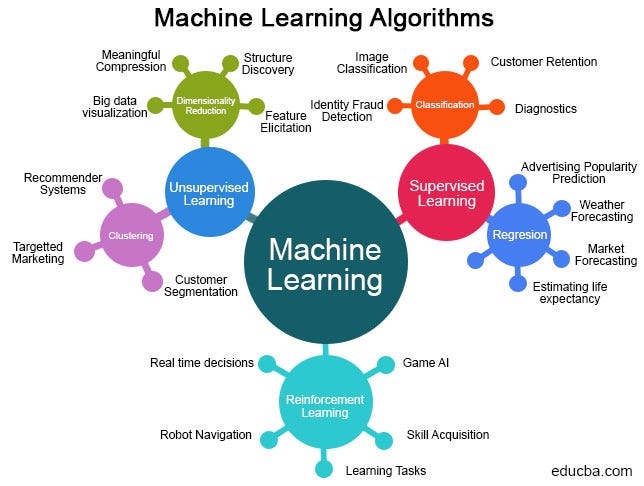
# ***Machine Learning***

*Machine Learning* merupakan sebuah teknologi yang memungkinkan mesin untuk belajar dan membuat keputusan secara otomatis tanpa instruksi pengguna. *Machine Learning* melalui tahap pembelajaran menggunakan data pelatihan khusus untuk mengotomatiskan proses pembuatan model analitis, yang memungkinkan model untuk membuat keputusan yang tepat dan menyelesaikan tugas-tugas terkait (Janiesch et al., 2021). Terutama pada tugas yang berkaitan dengan data *high-dimensional*, seperti klasifikasi, *clustering*, dan regresi, *Machine Learning* menunjukkan penerapan model yang baik. Teknologi ini digunakan dalam berbagai bidang, seperti pendidikan, ekonomi, teknologi, sosial, dan lainnya. Sudah banyak penerapan algoritma *machine learning* yang telah dilakukan seperti deteksi penipuan (Gupta et al., 2022), pengenalan ucapan dan gambar (Jimenez-Mesa et al., 2023), pemrosesan bahasa alami (NLP), dan lainnya (Janiesch et al., 2021).

Terdapat tiga tipe utama *Machine Learning* yang umum digunakan seperti pada **Gambar 4**, yaitu *Supervised Learning*, *Unsupervised Learning*, dan *Reinforcement Learning*. *Supervised Learning* adalah tipe *Machine Learning* di mana algoritma dilatih menggunakan kumpulan data yang diberi label, dengan setiap titik data memiliki label yang sesuai (Mobarak et al., 2023). Hal ini menyiratkan bahwa beberapa data telah ditandai dengan jawaban yang benar. Tipe ini diterapkan untuk model klasifikasi dan regresi. Beberapa algoritma yang umum digunakan dalam *Supervised Learning*, yaitu *Support Vector Machine* (SVM), Naïve Bayes, K-NN, *Random Forest*, *Linear Regression*, De*cision Tree*, dan lainnya (Allenbrand, 2023).

Berbeda dengan *Supervised Learning*, *Unsupervised Learning* tidak memerlukan data pelatihan yang diberi label. Tipe ini menganalisis serta mempelajari struktur dan pola secara ekslusif dari data yang tidak berlabel dan sudah ada sebelumnya, kemudian mengungkap informasi tersembunyi dari data yang diberikan (Mobarak et al., 2023). Tipe ini diterapkan untuk model *clustering*, deteksi anomali, dan reduksi dimensi (Zipfel et al., 2023). Berapa algoritma yang digunakan dalam *Unsupervised Learning*, yaitu K-Means, Fuzzy, C-Means, dan lainnya.

*Reinforcement Learning* adalah tipe *Machine Learning* di mana agen belajar berinteraksi dengan lingkungannya melalui *trial and* *error*, lalu menerima umpan balik dalam bentuk *reward* atau *punishment*. Umpan balik digunakan untuk menyesuaikan perilaku agen dan meningkatkan kinerjanya seiring waktu, Tujuan dari pembelajaran ini adalah mempelajari bagaimana mengambil tindakan untuk memaksimalkan *reward* yang didapatkan (Pichka, 2023). *Reinforcement Learning* telah diterapkan dalam berbagai bidang, seperti robotik, NLP, *game*, dan lainnya (Mandlekar et al., 2019) (H. Wang, 2021).



**Gambar 4.** Tipe dan Algoritma Machine Learning

# ***Oversampling***

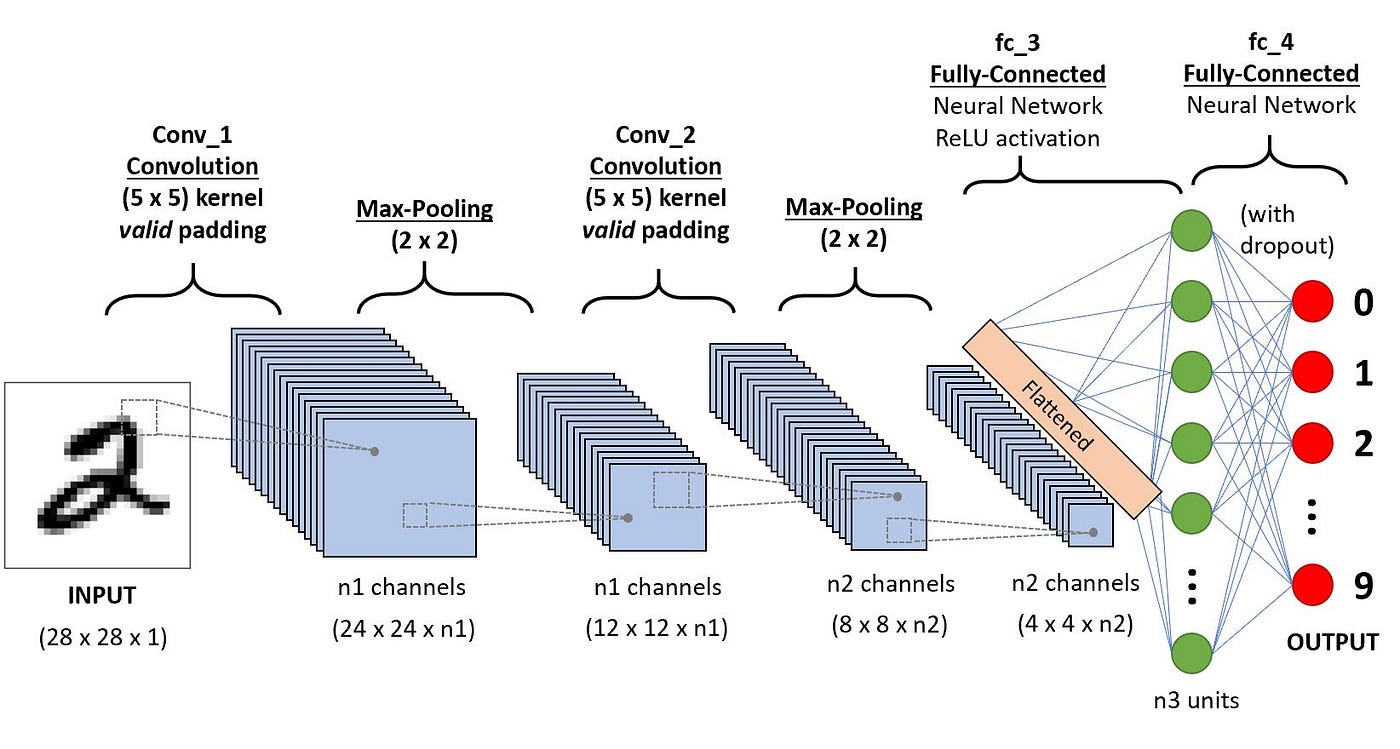
*Oversampling* merupakan sebuah teknik dalam *machine learning* untuk mengatasi permasalahan *imbalanced data.* Kondisi *imbalanced* *data* dapat memengaruhi kinerja model di mana hasil evaluasi menjadi bias. Hal ini dapat terjadi karena data pada kelas mayoritas yang sangat banyak, sedangkan pada kelas minoritas sangat sedikit, atau kedua kelas muncul secara bersamaan (Triyanto & Kusumaningrum, 2017). Teknik *oversampling* menyeimbangkan dataset dengan meningkatkan ukuran sampel dari kelas minoritas. Peningkatan ukuran sampel dilakukan dengan cara membuat data buatan sebanyak yang diperlukan dari kelas minoritas, sehingga jumlah sampel dari kelas minoritas menjadi seimbang dengan kelas mayoritas (Qadrini et al., 2022). Salah satu metode *oversampling* yang umum digunakan adalah *Synthetic Minority Over-sampling Technique* (SMOTE).

Metode SMOTE mampu menangani permasalahan *imbalanced class* pada dataset. Metode ini menghasilkan data sintetis dengan menerapkan interpolasi linier antara titik kelas minoritas dan salah satu kelas minoritas tetangga terdekat (Elreedy et al., 2023). Tetangga dari tetangga terdekat dipilih secara acak, tergantung pada jumlah *oversampling* yang diperlukan. Sampel sintetis dihasilkan dengan cara mengambil selisih antara vektor fitur (sampel) yang dipertimbangkan dengan tetangga terdekatnya. Kemudian, hasil selisih dikalikan dengan angka acak antara 0 dan 1, dan tambahkan ke vektor fitur yang sedang pertimbangkan. Hal ini menyebabkan pemilihan titik acak di sepanjang segmen garis di antara dua fitur tertentu (Chawla et al., 2002). Berikut *pseudocode* dari algoritma SMOTE.

|  |
| --- |
| **Algorithm** SMOTE(T, N, k)  **Input**: Jumlah sampel kelas minoritas T; Jumlah SMOTE N%; Jumlah tetangga  terdekat *k*  **Output**: (N/100)\* Sampel kelas minoritas sintetis T   1. (∗ *If N is less than 100%, randomize the minority class samples as only a random percent of them will be SMOTEd.* ∗) 2. **if** N <100 3. **then** Mengacak sampel kelas minoritas T 4. T = (N/100) ∗ T 5. N = 100 6. **endif** 7. N = (int)(N/100)( ∗ *Jumlah SMOTE diasumsikan dalam kelipatan integral 100*. ∗) 8. *k* = Jumlah tetangga terdekat 9. *numattrs* = Jumlah atribut 10. *Sample*[ ][ ]: array untuk sampel kelas minoritas asli 11. *newindex*: terus menghitung jumlah sampel sintetis yang dihasilkan, diinisialisasi ke 0 12. *Synthetic*[ ][ ]: Array untuk sampel sintetis   (∗ *Hitung k tetangga terdekat hanya untuk setiap sampel kelas minoritas.* ∗)   1. **for** i ← 1 **to** T 2. Hitung *k* tetangga terdekat untuk i, dan simpan indeks di *nnarray* 3. Populate(N, *i, nnarray*) 4. **endfor**   *Populate*(*N, i, nnarray*) (∗ *Fungsi untuk menghasilkan sampel sintetis.* ∗)   1. **while** *N* 0 2. Pilih angka acak antara 1 dan *k*, sebut saja *nn*. Langkah ini memilih   Salah satu *k* tetangga terdekat dari *i*.   1. **for** *attr* ← 1 **to** *numattrs* 2. Compute: *dif* = *Sample*[*nnarray*[*nn*]][*attr*] − *Sample*[*i*][*attr*] 3. Compute: *gap* = angka acak antara 0 dan 1 4. *Synthetic*[*newindex*][*attr*] = *Sample*[*i*][*attr*] + *gap* ∗ *dif* 5. **endfor** 6. *newindex*++ 7. *N = N* – 1 8. **endwhile** 9. **return** (∗ *End of Populate*. ∗)   End of Pseudo-Code. |

# ***Convolutional Neural Network* (CNN)**

*Convolutional Neural Network* (CNN) merupakan salah satu algoritma *deep learning* yang termasuk dalam jaringan *feed forward* di mana informasi mengalir searah dari *input* ke *output*. CNN dirancang untuk tugas pengolahan citra baik gambar, ucapan, atau sinyal audio, seperti klasifikasi, deteksi, segmentasi, dan prediksi (Keita, 2023). Pada beberapa penelitian, CNN juga digunakan pada data *time series* (Kim et al., 2023) dan *spatio-temporal* (L. Zhang et al., 2020). CNN terdiri dari lapisan-lapisan utama, yaitu *convolutional layer*, *ReLU layer*, *pooling layer*, dan *fully connected layer.* Arsitektur CNN secara umum (Keita, 2023) dapat dilihat pada **Gambar 5**.



**Gambar 5.** Arsitektur Convolutional Neural Network (CNN)

*Input* pada *convolutional layer* adalah sebuah gambar dengan ukuran di mana adalah tinggi dan lebar pada gambar dan adalah jumlah saluran atau kedalaman (sesuai dengan RGB dalam gambar, jika hitam putih = 1 dan berwarna = 3). Pada *convolutional layer*, juga terdapat pendeteksi fitur yang dikenal sebagai kernel atau filter dengan ukuran di mana lebih kecil dibandingkan dimensi gambar dan dapat sama dengan jumlah saluran atau lebih kecil dan mungkin berbeda pada tiap kernel. Kernel akan memeriksa apakah fitur tersebut ada pada gambar dengan melintasi bidang reseptif gambar, yang disebut dengan konvolusi. Operasi konvolusi dilakukan berulang kali hingga kernel mencakup seluruh bagian gambar dengan ukuran filter berupa matriks 3x3. Hasil proses konvolusi disebut *feature map* yang dapat dilihat pada **Gambar 6**.



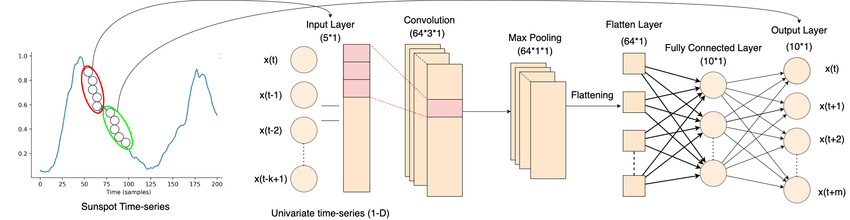
**Gambar 6.** Proses Konvolusi pada Convolutional Layer

*Pooling layer* bertujuan untuk mengurangi jumlah parameter (*spatial resolution*) pada *feature map* yang disebut operasi *spatial pooling*. Pada *layer* ini, kernel mengisi *output* *array* dengan menerapkan fungsi agregasi dalam bidang reseptif. Terdapat dua jenis *spatial pooling*, yaitu *average pooling* dan *max pooling*. Perbedaan kedua jenis *pooling* ini ditunjukan pada **Gambar 7**. Matriks dari *feature map* dibagi-bagi dengan ukuran . Pada *average pooling*, filter menghitung nilai rata-rata untuk dikirim ke *output array*. Sedangkan pada *max pooling*, filter memilih nilai maksimum dari tiap bagian untuk dikirim ke *output array*. Ekstraksi fitur ini mengurangi risiko *overfitting* dan kompleksitas, serta meningkatkan efisiensi model.



**Gambar 7.** Proses Spatial Pooling pada Pooling Layer

Pada *fully connected layer*, setiap node pada *output layer* terhubung langsung ke node sebelumnya dengan *input* matriks satu dimensi. CNN menerapkan fungsi aktivasi ReLU (*Rectified Linear Unit)* untuk non-linieritas pada model. Lapisan ini melakukan tugas seperti klasifikasi dan regresi berdasarkan fitur yang telah diekstrak sebelumnya menggunakan *softmax*. Selain data citra, CNN juga digunakan untuk melakukan klasifikasi pada data *time series* (Zheng et al., 2014)(Lara-Benítez et al., 2020) dengan arsitektur yang ditunjukkan pada **Gambar 8**.



**Gambar 8.** Arsitektur Convolutional Neural Network untuk Time Series

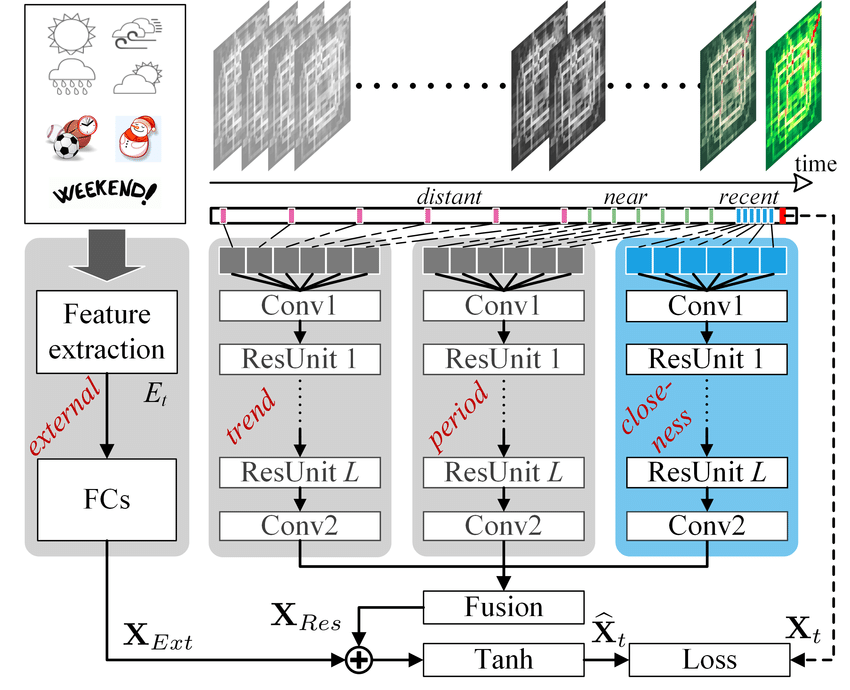
*Input layer* berisi masukan data *time series* berukuran , di mana adalah jumlah data yang menjadi masukan dan adalah panjang data. *Convolutional layer* berisi filter berukuran , di mana adalah jumlah filter dan adalah ukuran filter dengan merupakan panjang filter dan merupakan jumlah data dari *input layer*. Nilai dan adalah parameter yang dinamis.

*Pooling layer* sama seperti arsitektur CNN secara umum, yaitu *max pooling* dan *average pooling*. *Flatten layer* merupakan lapisan di mana hasil operasi konvolusi dan *spatial pooling* disambung menjadi sebuah *time-series* baru dan menjadi input untuk *fully connected layer*. *Output layer* merupakan lapisan untuk melakukan tugas klasifikasi *time series* yang memiliki buah neuron, di mana adalah jumlah kelas untuk klasifikasi terkait.

# ***Spatio-Temporal Residual Network* (ST-ResNet)**

ST-ResNet atau *Spatio-Temporal* *Residual Network* merupakan jenis arsitektur *Convolutional Neural Network* (CNN) yang digunakan untuk mengembangkan data spasial-temporal. Arsitektur ini dikombinasikan dari dua konsep utama, yaitu *Residual Network* (ResNet) dan pemrosesan data spasial-temporal. ResNet adalah jenis arsitektur CNN yang menggunakan model yang sudah dilatih sebelumnya dan mampu mengatasi masalah *vanishing gradient* dengan *residual connections* (*shortcut connections* atau *skip connections*) (Basodi et al., 2020). ResNet memiliki keuntungan, seperti memerlukan waktu pelatihan yang singkat dan mampu mengurangi risiko *overfitting* (Ridhovan & Suharso, 2022).

ST-ResNet dapat digunakan untuk memodelkan dan memahami hubungan temporal dan spasial dalam data seiring waktu. Hal ini sangat berguna dalam konteks di mana data berubah sepanjang waktu, seperti data citra waktu atau rangkaian waktu spasial-temporal. Arsitektur ST-ResNet terbagi menjadi empat komponen utama, yaitu *external, trend, period,* dan *closeness* (J. Zhang et al., 2017). Berikut adalah aristektur ST-ResNet yang ditunjukkan dalam **Gambar 9**.



**Gambar 9.** Arsitektur ST-ResNet. Conv: *Convolution*; ResUnit: *Residual Unit*; FC: *Fully-connected*.

Pada tahap konvolusi, konvolusi *multi-layer* digunakan untuk mengukur hubungan antar wilayah yang tidak berdekatan. Tiga komponen periodik (*trend, period, closeness*) dan komponen eksternal digabungkan melalui metode fusi berbasi parametrik-matriks. Berikut *pseudocode* dari proses pelatihanarsitektur ST-ResNet.

**Algorithm** ST-ResNet

**Input**: Historical observations: {X0, · · · ,X*n−*1};

external features: {*E0*, · · ·,*En−*1};

lengths of *closeness, period, trend* sequences: *lc*, *lp*, *lq*;

peroid: *p*; trend span: *q*.

**Output**: Learned ST-ResNet model

// *construct training instances*

1. D ←− ∅
2. **for** *all available time interval* t(1 ≤ t ≤ *n* − 1) **do**
3. *Sc* = [X*t−lc* ,X*t−(lc−*1*)*, · · · ,X*t−*1]
4. *Sp* = [X*t−lp·p*,X*t−(lp−1)·p*, · · · ,X*t−p*]
5. *Sq* = [X*t−lq·q*,X*t−(lq−1)·q*, · · · ,X*t−q*]

// *Xt is the target at time t*

1. put an training instance ({*Sc, Sp, Sq,Et*},X*t*) into *D*

// train the model

1. initialize all learnable parameters *θ* in ST-ResNet
2. **repeat**
3. randomly select a batch of instances *Db* from *D*
4. find *θ* by minimizing the objective (6) with *Db*
5. **until** *stopping criteria is met*

# **Evaluasi Model**

# **Mean Squared Error (MSE)**

*Mean Squared Error* (MSE) adalah metrik yang digunakan untuk mengukur rata-rata kuadrat kesalahan antara nilai estimasi dan nilai aktual dalam statistik dan *machine learning*. Metrik ini biasanya digunakan dalam konteks analisis regresi untuk menilai kualitas prediksi model (Stewart, 2023). Rumus perhitungan MSE adalah sebagai berikut:

(2.2)

Keterangan:

*yi* = Nilai aktual ke-i

*pi* = Nilai hasil peramalan/prediksi ke-i

*n* = Banyaknya data / ukuran sampel

MSE yang lebih rendah menunjukkan bahwa prediksi model lebih dekat dengan nilai aktual, sementara MSE yang lebih tinggi menunjukkan dispersi yang lebih besar dari titik data di sekitar rata-rata. Oleh karena itu, MSE yang lebih kecil umumnya lebih disukai karena menunjukkan kesesuaian model yang lebih baik dengan data.

1. **Mean Absolute Error (MAE)**

*Mean Absolute Error* (MAE) adalah ukuran rata-rata kesalahan dalam kumpulan prediksi, tanpa memperhitungkan arahnya. Ini dihitung sebagai perbedaan absolut rata-rata antara nilai prediksi dan nilai aktual. Rumus perhitungan MAE adalah sebagai berikut:

(2.3)

Keterangan:

yi = Nilai aktual ke-i

pi = Nilai hasil peramalan/prediksi ke-i

n = Banyaknya data / ukuran sampel

Dalam MAE, *error* yang berbeda tidak ditimbang lebih atau kurang, tetapi skor meningkat secara linier dengan peningkatan *error*. Skor MAE diukur sebagai rata-rata nilai kesalahan absolut. Absolut adalah fungsi matematika yang membuat angka positif. Oleh karena itu, perbedaan antara nilai yang diharapkan dan nilai yang diprediksi bisa positif atau negatif dan tentu akan selalu positif ketika menghitung MAE (Schneider & Xhafa, 2022).

# **Studi Literatur**

| **No** | **Penulis** | **Judul** | **Tahun** | **Metode** | **Hasil** |
| --- | --- | --- | --- | --- | --- |
| 1. | Z. Wang et al. | *Air Quality Measurement Based on*  *Double-Channel Convolutional Neural Network Ensemble Learning* | 2019 | *Double-Channel Convolutional Neural Network* + *Weighted Feature Fusion* | Ekstraksi fitur dilakukan menggunakan setiap saluran (*upper* dan *lower*) untuk melatih gambar. Kemudian, menerapkan metode pembelajaran mandiri untuk memberi bobot dan menggabungkan vektor fitur yang telah diekstrak untuk mengukur kualitas udara. Metode ini memperoleh akurasi sebesar 87%. |
| 2. | Q. Zhang et al. | *Deep-AIR: A Hybrid CNN-LSTM Framework for*  *Fine-Grained Air Pollution Forecast* | 2020 | AirRes CNN + LSTM | Metode ekstraksi fitur spasial menggunakan arsitektur AirRes berpengaruh sangat baik pada akurasi prediksi kualitas udara yang mencapai 80%. Berdasarkan hasil peramalan zat polutan yang berbeda, model memberikan performa yang baik dalam meramalkan zat CO, tetapi kurang memuaskan pada zat O3. Hal tersebut dikarenakan kurangnya variasi data zat O3. |
| 3. | Kalajdjieski et al. | *Air Pollution Prediction with Multi-Modal Data and Deep Neural Networks* | 2020 | ResNet CNN + *Inception* V3 *pre-trained model* | Model pada penelitian ini hanya menggunakan data gambar dan informasi cuaca. *Inception* V3 digunakan untuk mengidentifikasi fitur tingkat rendah dan dikombinasikan dengan arsitektur ResNet yang menghasilkan akurasi prediksi AQI sebesar 88%. |
| 4. | Cheng et al. | *Stacked ResNet-LSTM and CORAL model for Multi-Site Air Quality*  *Prediction* | 2022 | ResNet CNN + LSTM + *model stacking strategy + Correlation Alignment* (CORAL) | Prediksi konsentrasi PM2.5 menggunakan model *stacked* ResNet-LSTM berdasarkan data historis kualitas udara dan data meteorologi menunjukkan hasil akurasi sebesar 80%. Kemudian, CORAL digunakan untuk mengatasi masalah ketersediaan data yang tidak mencukupi dan mampu meningkatkan prediksi PM2.5. |
| 5. | Ong et al. | *Dynamically Pre-Trained Deep Recurrent Neural Networks Using Environmental Monitoring Data for Predicting PM2.5* | 2016 | *Deep Recurrent Neural Network* + DynPT + ElasticNet | DRNN mampu mengekstrak informasi penting dari data yang tidak 100% lengkap dan akurat, serta cukup kuat untuk menangani *noise* dan *error*. Prediksi *time series* menggunakan DynPT memperoleh akurasi sebesar 69%. Sedangkan, prediksi PM2.5 memperoleh akurasi sebesar 63%. |
| 6. | Portal-Porras et al. | *Hybrid LSTM+CNN Architecture for Unsteady Flow Prediction* | 2023 | CNN + LSTM | LSTM digunakan untuk memprediksi koefisien lift untuk setiap *timestep*. Kemudian, berdasarkan prediksi tersebut, CNN memprediksi kecepatan dan tekanan. Metode ini menghasilkan prediksi 192.4 kali lebih cepat dibandingkan simulasi CFD . |
| 7. | Song et al. | *ResNet-LSTM for Real-Time PM2.5 and PM10 Estimation Using Sequential Smartphone Images* | 2020 | ResNet CNN + LSTM | Ekstraksi fitur spasial-temporal dari data gambar menggunakan ResNet18 memperoleh hasil akurasi prediksi PM2.5 sebesar 93% dan PM10 sebesar 87%. |