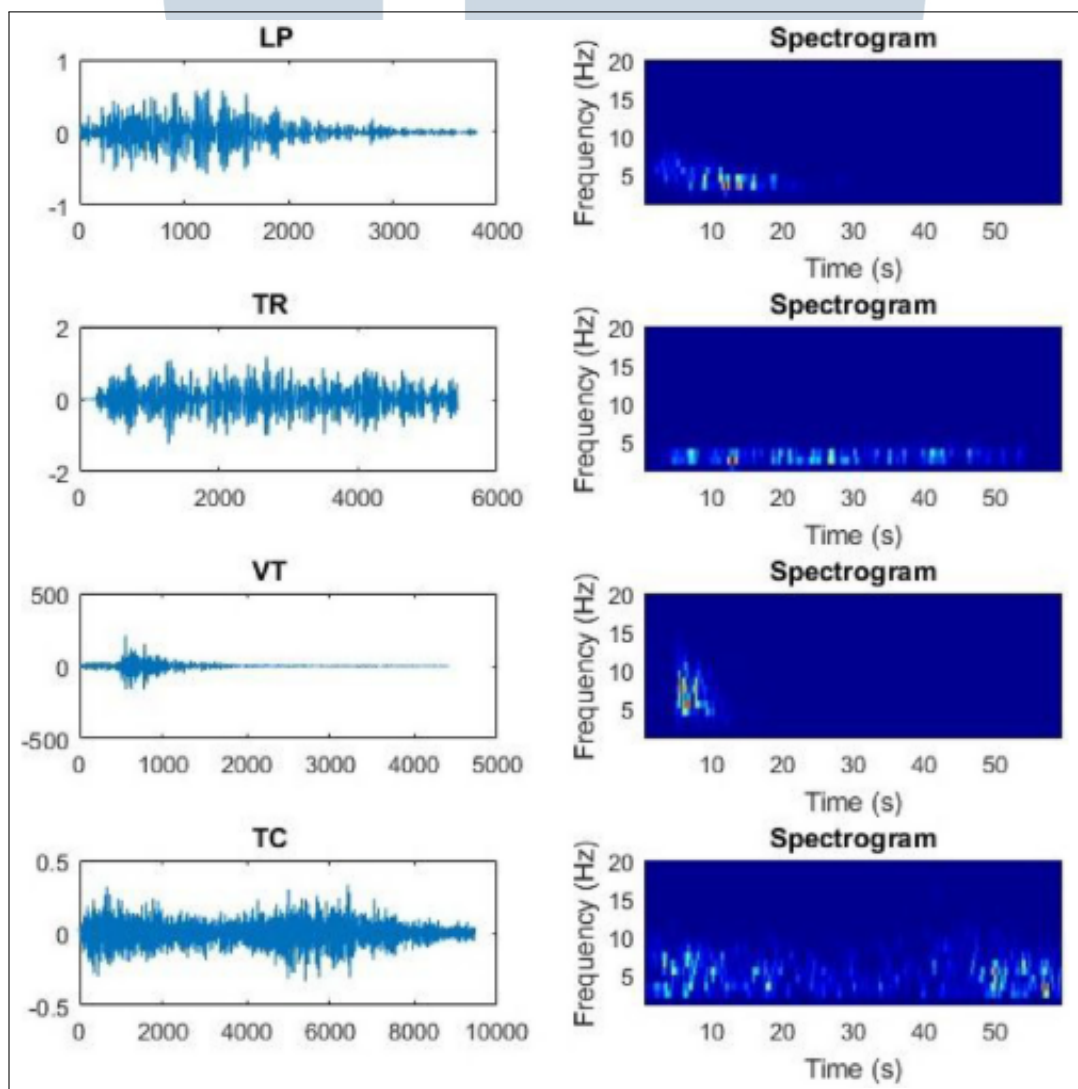


BAB 2

LANDASAN TEORI

2.1 Karakteristik Data Seismik Gunung Api

Data seismik memiliki beberapa karakteristik yang dapat dianalisis melalui *time domain* dalam *waveform* untuk mengamati perubahan amplitudo sinyal terhadap waktu dan *time-frequency domain* dalam bentuk spektrogram untuk mengamati perubahan kandungan frekuensi terhadap waktu. Contoh visualisasi dari kedua karakteristik tersebut ditunjukkan pada Gambar 2.1.



Gambar 2.1. Contoh Visualisasi Sinyal dan Spektrogram Jenis Gempa

Sumber: [19]

2.1.1 Data Seismik dalam Domain Waktu

Data seismik dalam domain waktu merupakan bentuk dasar data hasil perekaman secara kontinu yang diperoleh langsung dari seismometer, dikenal sebagai seismogram atau bentuk gelombang (*waveform*) [7]. Seismogram merupakan visualisasi yang menggambarkan perubahan nilai amplitudo seiring berjalannya waktu. Bentuk data ini menampilkan rekaman sinyal seismik mentah (*time-series*) menjadi bentuk gelombang satu dimensi yang diperoleh langsung dari stasiun pemantauan, sehingga tetap mempertahankan struktur temporal asli sinyal. Karakteristik sinyal dalam seismogram disajikan melalui tiga komponen utama. Sumbu horizontal (X) menunjukkan durasi/waktu kejadian, sumbu vertikal (Y) menunjukkan nilai amplitudo gelombang [7]. Sementara itu, komponen ketiga berupa morfologi gelombang merupakan karakteristik fisik sinyal, seperti pola awal kemunculan getaran (*onset*) dan bentuk gelombang.

Dalam vulkanologi, analisis domain waktu menjadi sangat penting karena aktivitas internal gunung api menghasilkan bentuk getaran yang khas berdasarkan jenis mekanisme sumber fisiknya. Karakteristik bentuk gelombang ini memberikan parameter visual yang jelas, sehingga beberapa jenis kejadian (*event*) gempa dapat langsung dikategorikan melalui pola umum gelombangnya. Melalui pendekatan ini, perbedaan karakteristik temporal seperti waktu tiba gelombang serta perubahan amplitudo dapat digunakan untuk mengidentifikasi jenis gempa secara cepat [20]. Analisis sinyal secara langsung tanpa transformasi tambahan memberikan keuntungan untuk mengenali karakteristik bentuk fisik asli dari setiap jenis gempa. Melalui domain waktu, karakteristik fisik gelombang dapat diamati secara langsung tanpa jeda waktu proses komputasi, yang pada akhirnya mempermudah proses klasifikasi awal dan pengenalan pola sinyal aktivitas vulkanik [17].

2.1.2 Data Seismik dalam Domain Waktu-Frekuensi

Selain domain waktu, data seismik juga dapat dianalisis dalam domain waktu-frekuensi menggunakan spektrogram. Spektrogram merupakan visualisasi yang menggambarkan perubahan frekuensi dari sinyal seiring berjalannya waktu. Spektrogram mengubah sinyal seismik mentah *time-series* menjadi bentuk citra dua dimensi yang kaya informasi melalui transformasi sinyal digital *Short-Time Fourier Transform (STFT)* [21]. Karakteristik sinyal dalam spektrogram disajikan melalui tiga komponen utama. Sumbu horizontal (X) menunjukkan waktu, sumbu vertikal

(Y) menunjukkan rentang frekuensi yang umumnya dinyatakan dalam satuan Hertz (Hz) [21]. Sementara itu, dimensi intensitas warna menunjukkan kekuatan energi atau magnitudo dari sinyal, warna terang menandakan energi yang kuat, sedangkan warna gelap menandakan energi yang minim.

Dalam vulkanologi, metode spektrogram menjadi penting karena sinyal yang dihasilkan bersifat kompleks dan non-stasioner, artinya karakteristik frekuensi sinyal terus berubah seiring waktu. Selain itu, sinyal-sinyal ini sering kali saling bertumpuk dan memiliki karakteristik yang serupa, sehingga beberapa jenis kejadian (*event*) gempa menjadi sulit dibedakan [22]. Akibat kompleksitas tersebut, data bentuk gelombang (*waveform*) sering kali belum cukup untuk mengidentifikasi jenis gempa secara akurat. Transformasi sinyal menjadi spektrogram memberikan keuntungan dalam mengekstrak karakteristik unik dari setiap jenis kejadian gempa dengan mengamati pola spektral yang lebih jelas dibandingkan *waveform* sehingga mempermudah analisis dan pengenalan pola sinyal setiap jenisnya [21].

Meskipun memiliki keunggulan tersebut, pendekatan ini memiliki keterbatasan dalam penerapannya. Keterbatasan pertama terletak pada potensi meningkatnya beban dan waktu komputasi karena data mentah harus ditransformasi terlebih dahulu menjadi citra spektrogram. Keterbatasan kedua adalah metode ini berpotensi menghilangkan informasi sinyal transien yang penting, seperti kepastian waktu kedatangan pertama (*onset*) Gelombang P (primer) dan S (sekunder), apabila parameter *windowing* transformasi tidak ditemukan dengan tepat [23].

2.2 Klasifikasi Kejadian Gempa

Kejadian (*event*) gempa merupakan salah satu indikator penting dalam pemantauan aktivitas gunung berapi karena mencerminkan berbagai aktivitas geofisika yang terjadi di dalam maupun di sekitar gunung, seperti rekahan batuan, pergerakan magma atau gas, resonansi fluida, serta aktivitas tektonik. Setiap jenis kejadian gempa memiliki karakteristik yang berbeda-beda seperti bentuk gelombang dan rentang frekuensi, masing-masing kejadian juga memberikan informasi mengenai mekanisme sumber yang berbeda sebagai indikasi aktivitas yang sedang terjadi di dalam gunung sehingga proses identifikasi dan klasifikasi kejadian gempa menjadi aspek penting untuk menilai aktivitas gunung berapi [17]. Adapun beberapa jenis kejadian (*event*) gempa dengan karakteristik dan mekanisme sumbernya yang dapat dilihat pada Tabel 2.1.

Tabel 2.1. Jenis Kejadian Gempa dan Karakteristiknya

Sumber: [17, 18]

Jenis Kejadian	Rentang Frekuensi	Karakteristik Sinyal	Mekanisme Sumber
VT (<i>Volcano-Tectonic</i>)	Mencapai 10 Hz	Sinyal <i>broadband</i> dengan <i>onset</i> impulsif menyerupai gempa tektonik.	Keretakan/Kehancuran pada batuan secara tiba-tiba akibat tekanan atau pergerakan fluida.
LP (<i>Long Period</i>)	0.5–5 Hz	Sinyal dengan spektrum frekuensi sempit yang didominasi frekuensi rendah.	Resonansi fluida (magma dan gas) di dalam saluran atau retakan.
TR (<i>Tremor</i>)	0.5–3 Hz	Sinyal kontinu dengan amplitudo relatif tinggi yang mengalami pelemahan secara bertahap pada akhir <i>event</i> .	Pergerakan magma berkelanjutan, resonansi fluida, atau kejadian seismik berulang dalam interval waktu yang rapat.
TC (<i>Tectonic</i>)	>5 Hz	Sinyal impulsif dengan pola peluruhan eksponensial dan karakteristik spektral menyerupai VT.	Aktivitas sesar geologi lokal, regional, atau aktivitas tektonik dari sumber yang jauh.

2.3 Short-Time Fourier Transform (STFT)

Short-Time Fourier Transform (STFT) merupakan metode transformasi sinyal dari domain waktu ke domain waktu-frekuensi dengan menerapkan Transformasi Fourier pada segmen-segmen sinyal berdurasi pendek. STFT digunakan untuk menganalisis sinyal nonstasioner karena mampu menampilkan perubahan karakteristik frekuensi sinyal terhadap waktu. Alur pengerjaan STFT dimulai dengan membagi sinyal masukan menjadi segmen-segmen pendek yang saling tumpang tindih (*overlap*), kemudian setiap segmen dikalikan dengan fungsi jendela (*window function*). Selanjutnya, algoritma *Fast Fourier Transform* (FFT) diterapkan pada setiap segmen yang telah diberi jendela untuk menghitung spektrum frekuensi secara efisien. Proses ini dilakukan secara berurutan pada setiap pergeseran waktu (*time frame*), sehingga diperoleh perubahan distribusi frekuensi sepanjang waktu.

Secara matematis, STFT dinyatakan pada Rumus 2.1 [24].

$$\text{STFT}(t, f) = \int_{-\infty}^{\infty} s(\tau) h^*(\tau - t) e^{-j2\pi f \tau} d\tau \quad (2.1)$$

Berdasarkan Rumus 2.1, $x(t)$ merupakan sinyal masukan pada domain waktu, $w(t - \tau)$ merupakan fungsi jendela (*window function*) yang digunakan untuk membatasi sinyal pada interval waktu tertentu, τ menyatakan posisi jendela terhadap waktu, f menyatakan frekuensi, sedangkan $\text{STFT}\{x(t)\}(\tau, f)$ merupakan hasil transformasi STFT yang menunjukkan distribusi frekuensi sinyal pada setiap posisi waktu τ [24].

2.4 Konversi Magnitudo ke Skala Desibel (dB)

Konversi magnitudo ke skala Desibel (dB) merupakan proses mengubah nilai amplitudo spektral ke dalam skala logaritmik. Konversi ini bertujuan agar pola energi pada spektrogram dapat terlihat lebih jelas. Selain itu, skala desibel membantu mempertegas karakteristik frekuensi dari setiap jenis kejadian gempa sehingga memudahkan perbedaan antara sinyal dan *noise* [22]. Secara matematis, konversi magnitudo ke skala desibel dinyatakan pada Rumus 2.2 [22].

$$dB = 20 \times \log_{10}(\text{Amplitudo}) \quad (2.2)$$

2.5 Normalisasi Min-Max Scaling

Min-Max Scaling merupakan metode normalisasi yang memetakan nilai data ke dalam rentang tertentu, umumnya 0 hingga 1 [25]. Pada citra, metode ini bertujuan untuk menghasilkan rentang nilai piksel yang konsisten sehingga memudahkan proses pembelajaran oleh model. Secara matematis, proses normalisasi *Min-Max Scaling* dinyatakan pada Rumus 2.3.

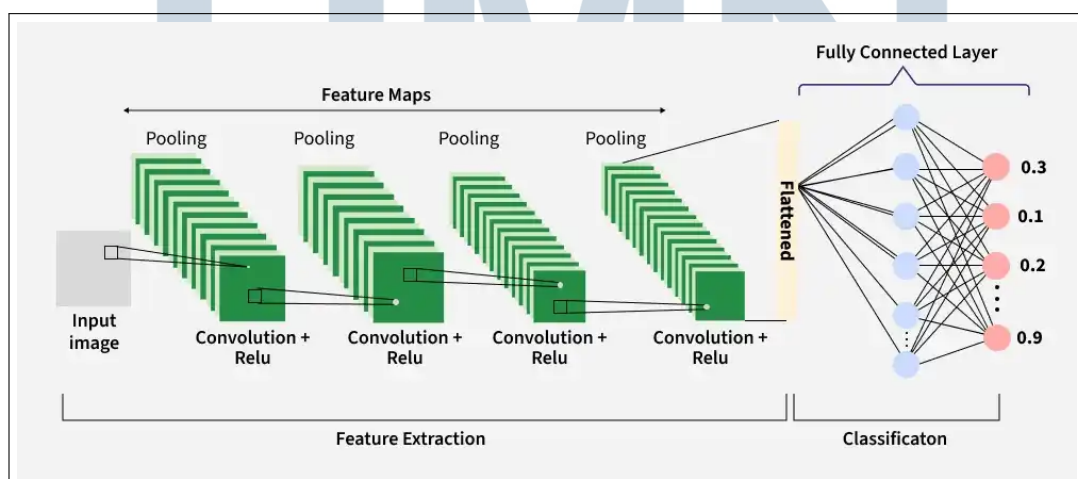
$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (2.3)$$

Berdasarkan Rumus 2.3, x merupakan nilai piksel sebelum normalisasi, x_{min} merupakan nilai piksel minimum, x_{max} merupakan nilai piksel maksimum, sedangkan x' merupakan nilai piksel setelah proses normalisasi menggunakan metode *Min-Max Scaling*.

2.6 CNN (Convolutional Neural Network)

Convolutional Neural Network atau CNN merupakan jenis jaringan saraf buatan yang dirancang untuk menangani data dengan struktur spasial, seperti citra (gambar & video). Berbeda dengan jaringan saraf konvensional, CNN menggunakan lapisan konvolusi untuk menangkap dan mengekstraksi fitur lokal secara otomatis serta hierarkis langsung dari data input [26]. Selain itu, CNN juga memiliki keunggulan dalam efisiensi komputasi melalui penerapan *local connectivity* dan *weight sharing* pada lapisan konvolusi, di mana filter yang sama digunakan secara berulang untuk memproses data input [25]. Mekanisme ini mampu mengurangi jumlah parameter yang harus dilatih secara signifikan, sehingga meningkatkan efisiensi model dan mempercepat proses pelatihan tanpa mengurangi kemampuan model dalam mempertahankan fitur-fitur penting dari data.

Arsitektur CNN pada umumnya terdiri dari dua blok utama, yaitu blok ekstraksi fitur (*feature extraction*) mencakup *convolutional layer* dan *pooling layer*, serta blok klasifikasi (*classification*) terdiri dari *fully connected layer*. Ilustrasi dari arsitektur CNN dapat dilihat pada Gambar 2.2.



Gambar 2.2. Arsitektur *Convolutional Neural Network* (CNN)

Sumber: [27]

2.6.1 Convolutional Layer

Lapisan konvolusi merupakan komponen utama arsitektur CNN yang berfungsi untuk mengekstraksi fitur dari data input melalui operasi konvolusi. Operasi ini dilakukan dengan mengalikan filter (*kernel*) dengan bagian lokal dari data input, kemudian menjumlahkan hasil perkaliannya untuk menghasilkan keluaran berupa *feature map* yang mewakili pola tertentu dari data [26]. Pada proses konvolusi terdapat beberapa parameter penting, yaitu *Stride* menentukan langkah perpindahan kernel ketika melintasi data, *padding* berfungsi mengurangi kehilangan informasi pada area tepi data input, dan jumlah filter menentukan banyaknya fitur yang dipelajari model [25]. Hasil konvolusi kemudian dilanjutkan dengan menerapkan fungsi aktivasi non-linear seperti ReLU (*Rectified Linear Unit*).

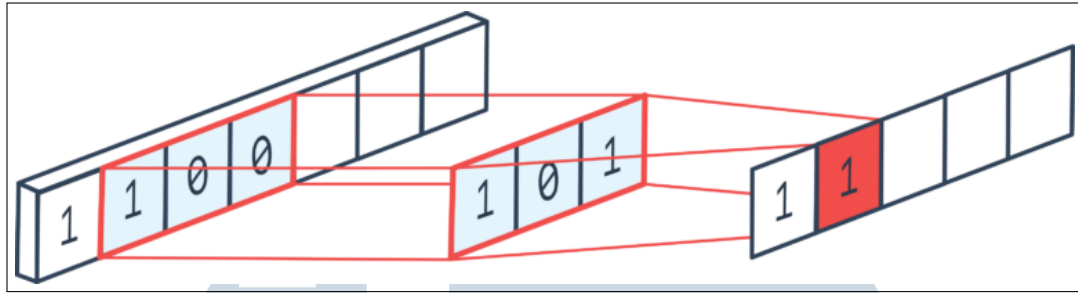
Penerapan operasi konvolusi dalam CNN dapat menyesuaikan dengan dimensi data input. Pada data satu dimensi, seperti data deret waktu (*time-series*) atau sekuensial, digunakan konvolusi satu dimensi (*1D convolution*), sedangkan pada data dua dimensi seperti citra digunakan konvolusi dua dimensi (*2D convolution*).

A Konvolusi 1 Dimensi (CNN 1D)

Konvolusi satu dimensi digunakan untuk memproses data yang memiliki struktur sekuensial, seperti deret waktu. Pada operasi ini, kernel bergerak sepanjang satu sumbu input untuk mengekstraksi pola lokal dari data. Pada setiap posisi, nilai-nilai input yang berada dalam cakupan kernel dikalikan dengan bobot kernel yang bersesuaian, kemudian hasilnya dijumlahkan untuk menghasilkan satu nilai keluaran [28]. Secara matematis, operasi konvolusi satu dimensi dinyatakan pada Rumus 2.4.

$$C(i) = \sum_{m=0}^{M-1} I(i+m) \cdot K(m) \quad (2.4)$$

Pada Rumus 2.4, $I(i)$ menyatakan nilai input pada posisi ke- i , $K(m)$ menyatakan nilai kernel pada posisi ke- m , M merupakan panjang kernel, sedangkan $C(i)$ merupakan nilai keluaran hasil konvolusi pada posisi ke- i . Hasil konvolusi yang diperoleh pada setiap pergeseran kernel membentuk *feature map* yang memuat informasi atau karakteristik penting dari data masukan. Ilustrasi operasi konvolusi satu dimensi dapat dilihat pada Gambar 2.3.



Gambar 2.3. Ilustrasi Proses Konvolusi Satu Dimensi (1D)

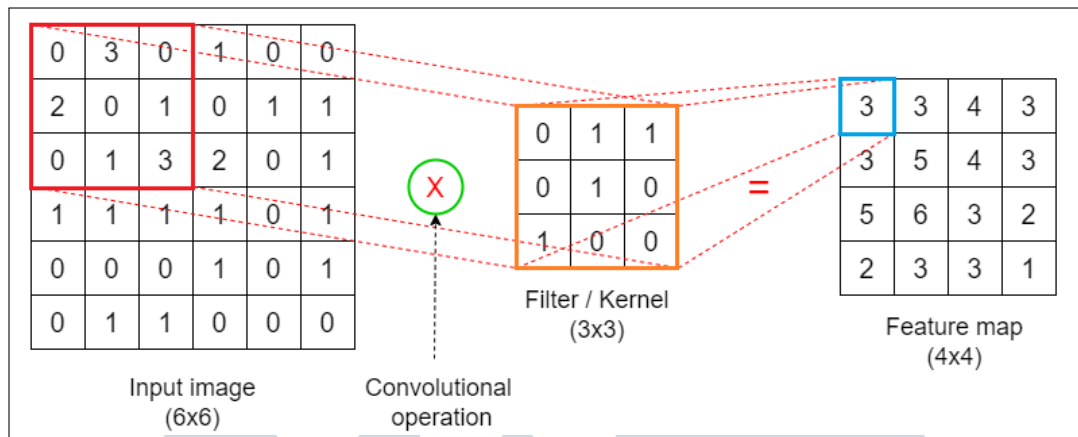
Sumber: [29]

B Konvolusi 2 Dimensi (CNN 2D)

Konvolusi dua dimensi digunakan untuk memproses data berbentuk matriks, seperti citra, di mana kernel bergerak secara horizontal dan vertikal untuk mengekstraksi hubungan spasial antar elemen input [26]. Pada setiap posisi, nilai-nilai input yang berada dalam cakupan kernel dikalikan dengan bobot kernel yang bersesuaian, kemudian hasilnya dijumlahkan untuk menghasilkan satu nilai keluaran. Secara matematis, operasi konvolusi dua dimensi dinyatakan pada Rumus 2.5.

$$C(i, j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(i+m, j+n) K(m, n) \quad (2.5)$$

Pada Rumus 2.5, $I(i, j)$ menyatakan nilai input pada posisi (i, j) , $K(m, n)$ menyatakan nilai kernel pada posisi (m, n) , M dan N merupakan ukuran kernel, sedangkan $C(i, j)$ merupakan nilai keluaran hasil konvolusi pada posisi (i, j) . Ilustrasi operasi konvolusi dua dimensi dapat dilihat pada Gambar 2.4, di mana kernel berukuran 3×3 diaplikasikan pada sebagian area input berukuran 6×6 sehingga menghasilkan *feature map*.



Gambar 2.4. Ilustrasi Proses Konvolusi Dua Dimensi (2D)

Sumber: [30]

2.6.2 Activation Function

Fungsi aktivasi merupakan komponen penting pada model *neural network* seperti CNN karena berfungsi memberikan sifat non-linearitas pada jaringan sehingga memungkinkan model mampu mempelajari pola yang kompleks dari data [31]. Tanpa fungsi aktivasi, model hanya mampu mempelajari pola linear pada data meskipun memiliki banyak lapisan. Terdapat beberapa jenis fungsi aktivasi dan masing-masing memiliki karakteristik dan kelebihan-kekurangan yang berbeda, beberapa fungsi aktivasi yang sering digunakan di antaranya *Hyperbolic Tangent (Tanh)*, *Sigmoid*, *Rectified Linear Unit (ReLU)*, dan *Softmax*.

A Rectified Linear Unit (ReLU)

Fungsi aktivasi *ReLU* merupakan salah satu fungsi aktivasi non-linear yang umum digunakan. Fungsi ini bekerja dengan mempertahankan nilai input positif dan mengubah nilai input negatif menjadi nol [26]. Fungsi aktivasi *ReLU* dinyatakan pada Rumus 2.6.

$$f(x) = \max(0, x) \quad (2.6)$$

Berdasarkan Rumus 2.6, x merupakan nilai input neuron, sedangkan $f(x)$ merupakan nilai keluaran hasil aktivasi. Dibandingkan fungsi *Sigmoid* dan *Tanh*, *ReLU* membantu mengurangi permasalahan *vanishing gradient*, sehingga proses pelatihan model dapat berlangsung lebih cepat dan stabil [26]. Namun demikian, *ReLU* memiliki kelemahan berupa *dying ReLU*, kondisi ketika neuron menjadi tidak aktif karena secara konsisten menghasilkan nilai 0 akibat menerima input negatif [26, 31].

B Softmax

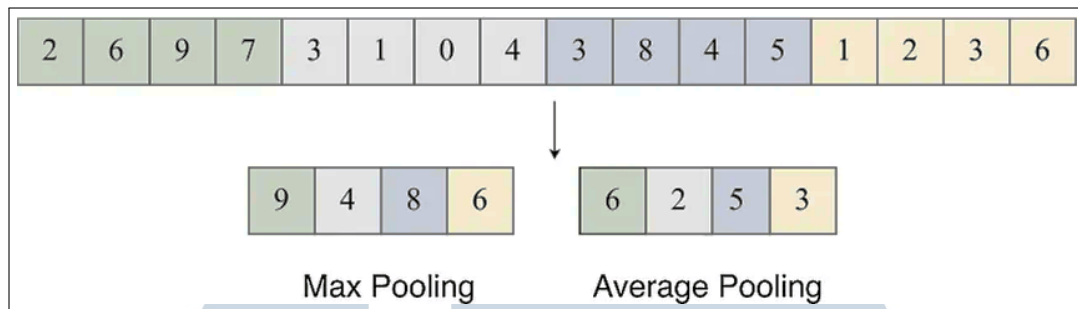
Fungsi aktivasi *Softmax* merupakan fungsi aktivasi yang mengubah nilai keluaran neuron menjadi distribusi probabilitas pada rentang antara 0 hingga 1 [26]. Hal ini memungkinkan model untuk menentukan probabilitas prediksi pada masing-masing kelas, sehingga fungsi aktivasi *Softmax* umum digunakan pada lapisan *output* untuk klasifikasi multikelas [26]. Fungsi aktivasi *Softmax* dinyatakan pada Rumus 2.7.

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (2.7)$$

Berdasarkan Rumus 2.7, z_i merupakan nilai *input* untuk kelas ke- i , K merupakan jumlah kelas, dan $\sigma(z_i)$ merupakan probabilitas keluaran untuk kelas ke- i . *Softmax* memiliki keterbatasan sensitif terhadap outlier, sehingga model dapat menghasilkan probabilitas yang terlalu dominan pada satu kelas tertentu [31].

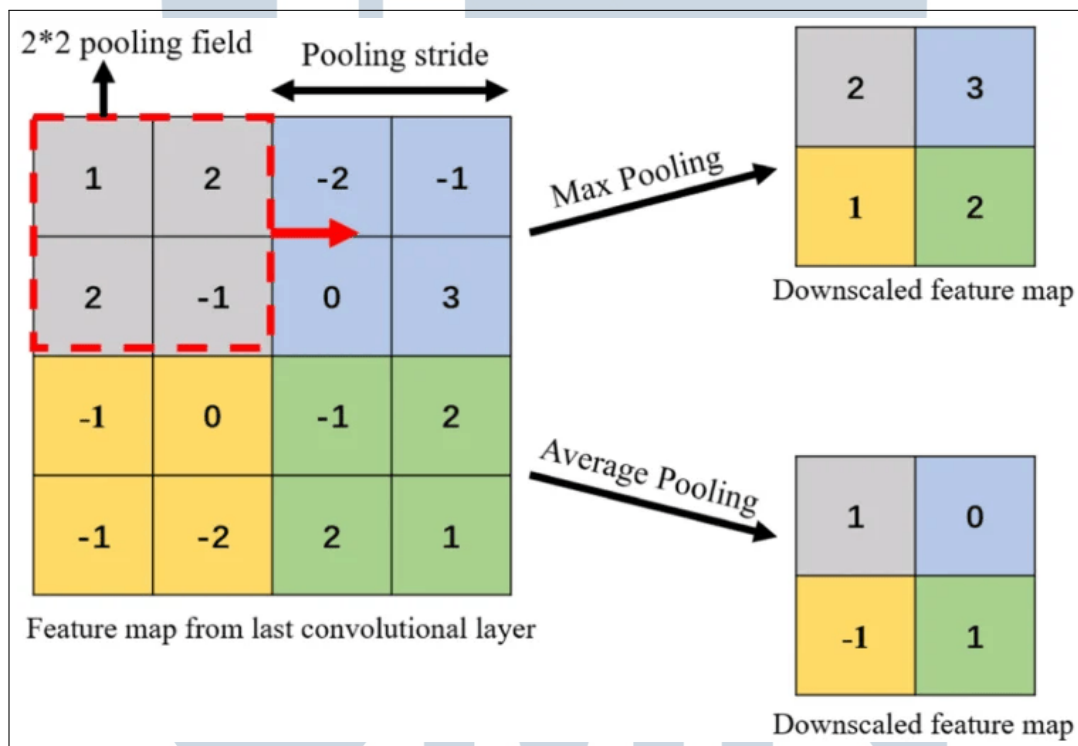
2.6.3 Pooling Layer

Lapisan *pooling* merupakan lapisan yang digunakan untuk mereduksi dimensi (*downsampling*) terhadap *feature map* hasil konvolusi. Proses ini bertujuan untuk mengurangi kompleksitas komputasi, menurunkan jumlah parameter, mempertahankan fitur penting hasil ekstraksi konvolusi, serta meningkatkan ketahanan model terhadap pergeseran kecil pada posisi fitur input [26, 32]. Secara umum, proses *pooling* dilakukan dengan menggeser sebuah *pooling window* pada area lokal *feature map*, kemudian melakukan operasi agregasi tertentu untuk menghasilkan keluaran dengan dimensi yang lebih kecil. Teknik *pooling* yang umum dilakukan adalah *max pooling* dan *average pooling* yang dapat dilihat ilustrasinya pada Gambar 2.5 untuk CNN 1D dan Gambar 2.6 untuk CNN 2D.



Gambar 2.5. Operasi *Max* dan *Average Pooling* CNN 1D

Sumber: [28]



Gambar 2.6. Operasi *Max* dan *Average Pooling* CNN 2D

Sumber: [32]

A Max Pooling

Metode *max pooling* memilih nilai maksimum dari seluruh elemen dalam suatu area *pooling window* sebagai keluaran [26]. Secara matematis, operasi *max pooling* dinyatakan pada Rumus 2.8.

$$f_{\max}(x) = \max\{x_i\}_{i=1}^N \quad (2.8)$$

Berdasarkan Rumus 2.8, x_i adalah elemen ke- i dalam *pooling window*, N adalah jumlah elemen dalam *pooling window*, dan $f_{\max}(x)$ adalah nilai keluaran hasil *max pooling*.

B Average Pooling

Metode *average pooling* menghitung nilai rata-rata dari seluruh elemen dalam suatu area *pooling window* dengan menjumlahkan seluruh elemen dalam area *pooling window*, kemudian dibagi dengan jumlah elemen [26]. Secara matematis, operasi *average pooling* dinyatakan pada Rumus 2.9.

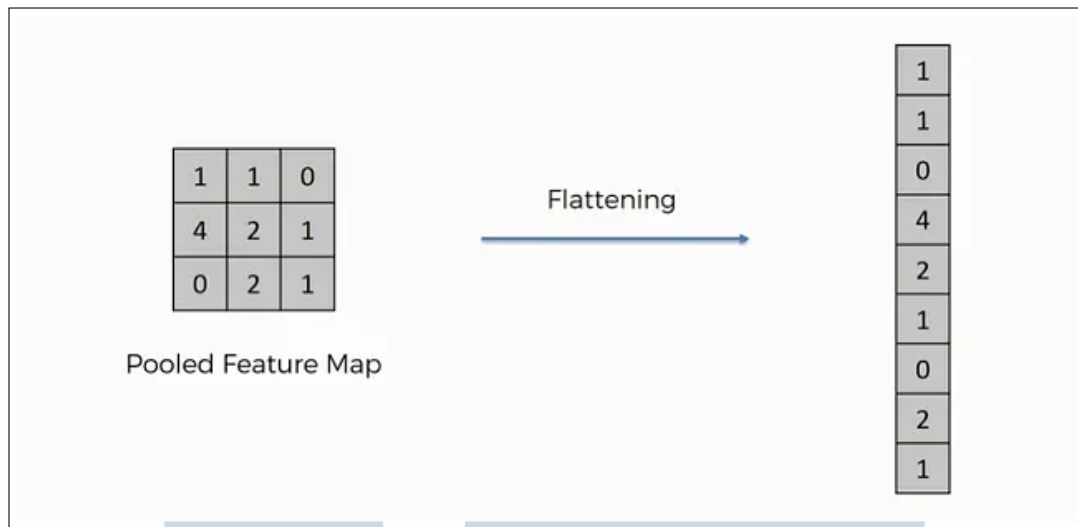
$$f_{\text{avg}}(x) = \frac{1}{N} \sum_{i=1}^N x_i \quad (2.9)$$

Berdasarkan Rumus 2.9, x_i adalah elemen ke- i dalam *pooling window*, N adalah jumlah elemen dalam *pooling window*, dan $f_{\text{avg}}(x)$ adalah nilai keluaran hasil *average pooling*.

2.6.4 Flattening Layer

Flatten mengubah *output* multidimensi dari *feature map* menjadi vektor satu dimensi sesuai dengan multiplikasi ukuran dimensi fitur sebelumnya sebelum memasuki lapisan *Fully Connected/Dense Layer* [26]. Ilustrasi proses flatten dapat dilihat pada Gambar 2.7.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 2.7. Ilustrasi Proses *Flattening*

Sumber: [33]

2.6.5 Fully Connected Layer

Fully Connected Layer atau disebut juga *Dense Layer* merupakan lapisan yang melakukan proses klasifikasi berdasarkan hasil *feature map* yang telah diekstrak oleh lapisan sebelumnya. Pada lapisan ini, setiap neuron terhubung dengan seluruh neuron pada lapisan terdahulu agar semua informasi fitur dapat dipertimbangkan dalam pengambilan keputusan [26]. Proses komputasi pada lapisan ini berjalan dengan menerima input berupa vektor satu dimensi hasil dari proses *flattening*. Setiap nilai input tersebut kemudian dikalikan dengan bobot (*weight*) masing-masing, lalu ditambah dengan nilai *bias*. Secara matematis, operasi *dense layer* ini dinyatakan pada Rumus 2.10.

$$y = \sum_{i=1}^n w_i x_i + b \quad (2.10)$$

x_i adalah nilai input ke- i , w_i adalah bobot yang terhubung dengan input ke- i , b adalah nilai *bias*, dan y adalah luaran (*output*) neuron. Luaran (*output*) dari *fully connected layer* kemudian diteruskan ke *output layer* untuk menghasilkan prediksi akhir menggunakan fungsi aktivasi yang sesuai dengan jenis permasalahan. Pada tugas klasifikasi multikelas, fungsi aktivasi yang umum digunakan adalah *Softmax* untuk menghasilkan nilai probabilitas di setiap kelas, sedangkan untuk tugas klasifikasi biner umumnya menggunakan fungsi aktivasi *Sigmoid*.

2.7 Penerapan CNN pada Data Seismik

Convolutional Neural Network (CNN) awalnya dirancang untuk menganalisis data dengan struktur spasial seperti citra digital, khususnya pada berbagai tugas *computer vision* salah satunya klasifikasi gambar. Namun seiring dengan perkembangannya, CNN juga diadaptasi untuk menangani data deret waktu (*time-series*) seperti sinyal, audio, data finansial, dan data cuaca melalui penerapan operasi konvolusi satu dimensi (*1D convolution*) untuk menangkap pola temporal lokal dalam urutan data [26]. Berkaitan dengan data seismik, pada dasarnya merupakan data rekaman dalam bentuk gelombang (*waveform*) sinyal yang secara struktural termasuk data deret waktu. Karakteristik data tersebut memungkinkan menerapkan CNN untuk analisis data seismik yang kompleks melalui dua pendekatan utama, yaitu berbasis sinyal menggunakan CNN 1D dan berbasis spektrogram menggunakan CNN 2D. Kedua pendekatan tersebut masing-masing memiliki metode ekstraksi fitur, kelebihan, dan kekurangan yang berbeda.

2.7.1 Studi Terdahulu Penerapan CNN pada Data Seismik

Penerapan *Convolutional Neural Network* (CNN) dalam analisis data seismik telah banyak diteliti untuk berbagai tugas, seperti deteksi dan klasifikasi jenis kejadian (*event*) seismik baik tektonik maupun vulkanik. Berbagai penelitian terdahulu menunjukkan bahwa CNN dapat diterapkan pada data seismik dengan pendekatan, jenis data masukan, serta tujuan analisis yang beragam, sesuai dengan karakteristik permasalahan yang diteliti. Ringkasan beberapa penelitian terdahulu terkait penerapan CNN pada analisis data seismik disajikan pada Tabel 2.3.

Tabel 2.3. Penelitian Terdahulu Penerapan CNN pada Analisis Data Seismik

Peneliti	Tugas Analisis	Input Data	Pendekatan	Temuan dan Hasil Utama
Fee et al. (2025) [22]	Klasifikasi multi-kelas <i>event</i> vulkanik	Spektrogram	VOISS-Net <i>Generalized</i> (CNN 2D)	Mencapai akurasi 87%, berhasil diuji pada data kontinu nyata multi gunung.
Lanjut ke halaman berikutnya				

Tabel 2.3. Penelitian Terdahulu Penerapan CNN pada Analisis Data Seismik (Lanjutan)

Peneliti	Tugas Analisis	Input Data	Pendekatan	Temuan dan Hasil Utama
Tan et al. (2024) [16]	Klasifikasi multi-kelas <i>event</i> vulkanik	Spektrogram	VOISS-Net (CNN 2D)	Akurasi mencapai 90%, berhasil mengklasifikasikan berbagai jenis tremor.
Lu et al. (2025) [34]	Klasifikasi gempa bumi, ledakan, dan runtuh	<i>Waveform</i> , spektrogram, fitur <i>P/S wave</i>	EQTypeNet	Akurasi tertinggi mencapai 99,5% dengan <i>transfer learning</i> .
Ferreira et al. (2023) [35]	Klasifikasi multi-kelas <i>event</i> vulkanik	Spektrogram	Multi-input CNN	Pendekatan multi-stasiun memberikan performa lebih baik dan efektif dibanding stasiun tunggal.
Yue et al. (2023) [36]	Klasifikasi gempa bumi, ledakan, runtuh	<i>Waveform</i> dan spektrum waktu–frekuensi	<i>Two-Step CNN</i>	Berhasil membedakan gempa alami, ledakan, dan runtuh dengan rata-rata akurasi 96,13%.
Manley et al. (2022) [17]	Klasifikasi multi-kelas <i>event</i> vulkanik	<i>Waveform</i>	SeismicNet (CNN 1D)	<i>Active learning</i> meningkatkan akurasi pengujian dengan data berlabel lebih sedikit.
Nakano et al. (2022) [37]	Klasifikasi gempa bumi, tremor, <i>noise</i> , dan erupsi	<i>Waveform</i> dan spektrum	CNN 1D dan CNN 2D	CNN 1D dan CNN 2D memberikan performa serupa, namun CNN 1D lebih sesuai untuk pemantauan <i>real-time</i> .
Soto et al. (2021) [38]	Deteksi fase gelombang P, tiga dan S	Seismogram	DeepPhasePick (CNN-RNN)	Berhasil melakukan deteksi fase gelombang (P dan S) secara akurat dan presisi.
Lanjut ke halaman berikutnya				

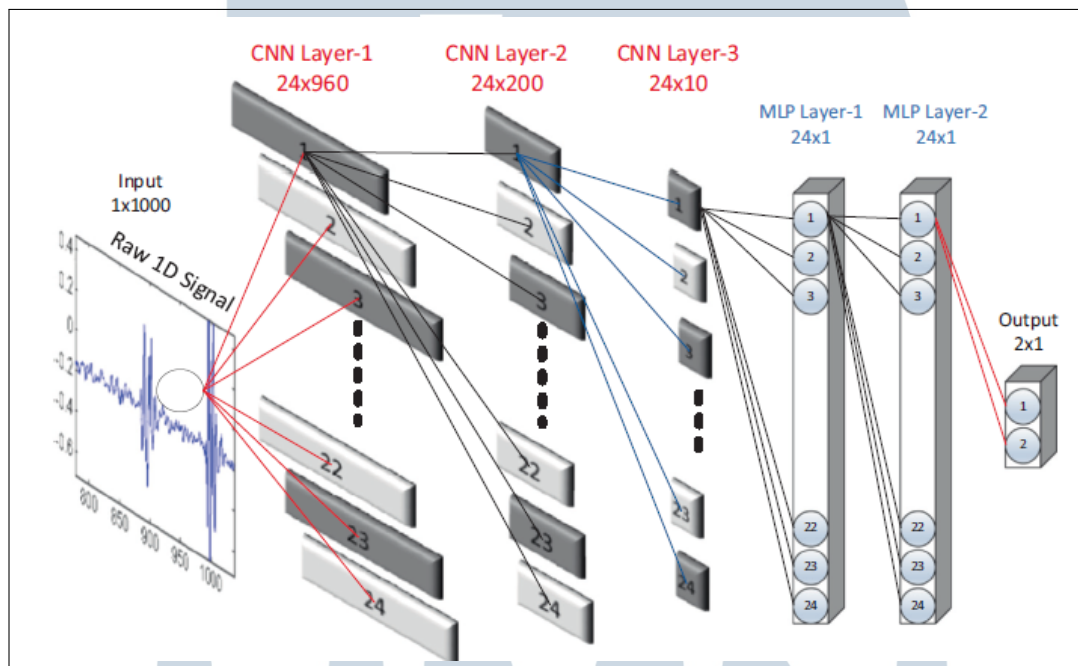
Tabel 2.3. Penelitian Terdahulu Penerapan CNN pada Analisis Data Seismik (Lanjutan)

Peneliti	Tugas Analisis	Input Data	Pendekatan	Temuan dan Hasil Utama
Takahashi et al. (2021) [39]	Klasifikasi gempa, tremor, dan <i>noise</i>	Spektrogram	CNN 2D	Berhasil mengklasifikasikan ketiga jenis sinyal dengan akurasi di atas 96%.
Ku et al. (2021) [40]	Deteksi dan klasifikasi <i>event</i> seismik	<i>Waveform</i> tiga komponen	<i>Attention-Based 1D CNN</i>	<i>Attention mechanism</i> meningkatkan akurasi deteksi dan klasifikasi hingga lebih dari 98%.
Canário et al. (2020) [18]	Klasifikasi multi-kelas <i>event</i> vulkanik	<i>Waveform</i> dan <i>scalogram</i>	SeismicNet (CNN 1D) dan CNN 2D	SeismicNet dengan <i>waveform</i> mencapai akurasi di atas 95% tanpa ekstraksi fitur manual dan transformasi sinyal ke domain waktu–frekuensi.
Calderón et al. (2020) [14]	Klasifikasi multi-kelas <i>event</i> vulkanik	Spektrogram	CNN 2D	CNN 2D dengan spektrogram mencapai akurasi 94,13% pada klasifikasi multi-kelas.
Salazar et al. (2020) [41]	Klasifikasi multikelas <i>event</i> vulkanik	Spektrogram	<i>Deep CNN</i>	<i>Deep CNN</i> mencapai akurasi 95% dengan AUC sebesar 0,98.

Berdasarkan Tabel 2.3 penelitian terdahulu di atas dapat disimpulkan bahwa CNN telah menunjukkan kemampuan yang baik dalam menangani berbagai tugas analisis data seismik. Selain itu, penelitian-penelitian tersebut menunjukkan bahwa CNN dapat memanfaatkan berbagai bentuk data masukan, baik berupa (*waveform*) sinyal secara langsung maupun data pada domain waktu–frekuensi seperti spektrogram dan skalogram. Perbedaan bentuk data masukan tersebut menghasilkan dua pendekatan utama dalam penerapan CNN pada data seismik, yaitu pendekatan berbasis *waveform* menggunakan CNN 1D dan pendekatan berbasis spektrogram menggunakan CNN 2D. Kedua pendekatan tersebut memiliki karakteristik yang berbeda dalam mempelajari informasi yang terkandung pada sinyal seismik.

2.7.2 CNN 1D untuk Analisis Waveform Seismik

Analisis data seismik dengan menerapkan CNN 1D adalah dengan mengolah langsung *waveform signal* dalam domain waktu sebagai *input* model, sehingga model mempelajari dan membedakan karakteristik temporal seperti amplitudo dan karakteristik bentuk fisik gelombang khas dari masing-masing jenis *event* gempa langsung dari sinyal, seperti yang ditunjukkan pada Gambar 2.8.



Gambar 2.8. Ilustrasi Proses CNN 1D Analisis Sinyal

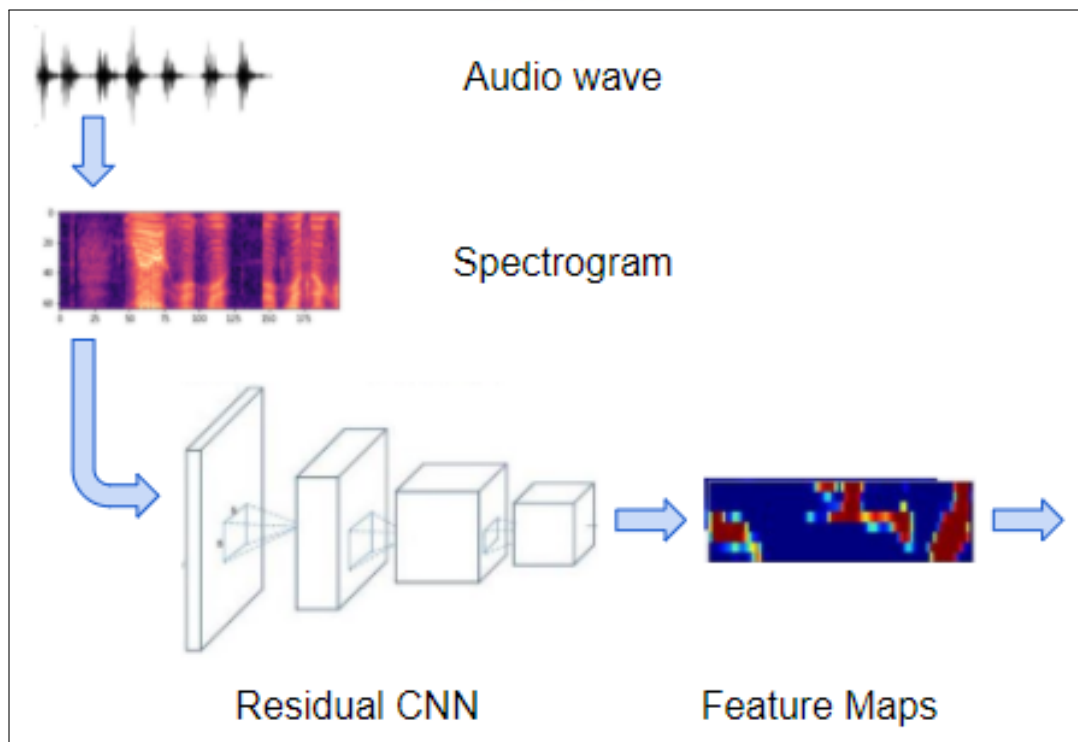
Sumber: [42]

Pendekatan dengan CNN 1D memiliki beberapa kelebihan jika dibandingkan dengan CNN 2D, di antaranya dapat memproses sinyal mentah secara langsung sehingga informasi temporal dari data mentah tetap terjaga dan komputasi yang lebih rendah karena kebutuhan pra-pemrosesan yang lebih sederhana tidak memerlukan transformasi data menjadi spektrogram, dan jumlah parameter model CNN 1D umumnya lebih rendah dibandingkan CNN 2D [18, 42]. Namun pendekatan ini juga memiliki keterbatasan ketika menghadapi karakteristik *waveform* antar gempa yang serupa sehingga memerlukan informasi spektral dari domain waktu-frekuensi, kemudian performa model juga dapat dipengaruhi kualitas sinyal seperti kandungan *noise* yang tinggi.

2.7.3 CNN 2D untuk Analisis Spektrogram

Analisis data seismik juga dapat dilakukan dengan menerapkan CNN 2D dengan memanfaatkan spektrogram sebagai *input* model. Proses diawali dengan mengubah sinyal domain waktu ke dalam domain waktu-frekuensi untuk menghasilkan data dua dimensi bernama spektrogram. Kemudian CNN 2D akan memproses spektrogram layaknya analisis citra dan akan mempelajari pola visual dari distribusi energi sinyal terhadap waktu dan frekuensi [26], sehingga model akan mengenali karakteristik seperti frekuensi dominan, perubahan pola frekuensi, distribusi energi spektral, hingga tekstur visual khas yang berkaitan dengan jenis kejadian seismik tertentu. Hal ini dapat dilihat lewat ilustrasi pada Gambar 2.9.

Kelebihan pendekatan CNN 2D adalah kemampuannya dalam menampilkan informasi karakteristik spektral yang kaya sehingga pola pembeda antar kelas yang sulit diamati pada *waveform* dapat dikenali dengan lebih baik [21]. Namun, pendekatan ini memiliki konsekuensi pada kebutuhan komputasi yang lebih tinggi karena jumlah parameter model CNN 2D yang umumnya cenderung lebih besar dan proses transformasi sinyal menjadi spektrogram, serta kebutuhan data latih yang besar dengan proses pelatihan umumnya lebih lama [34, 42].

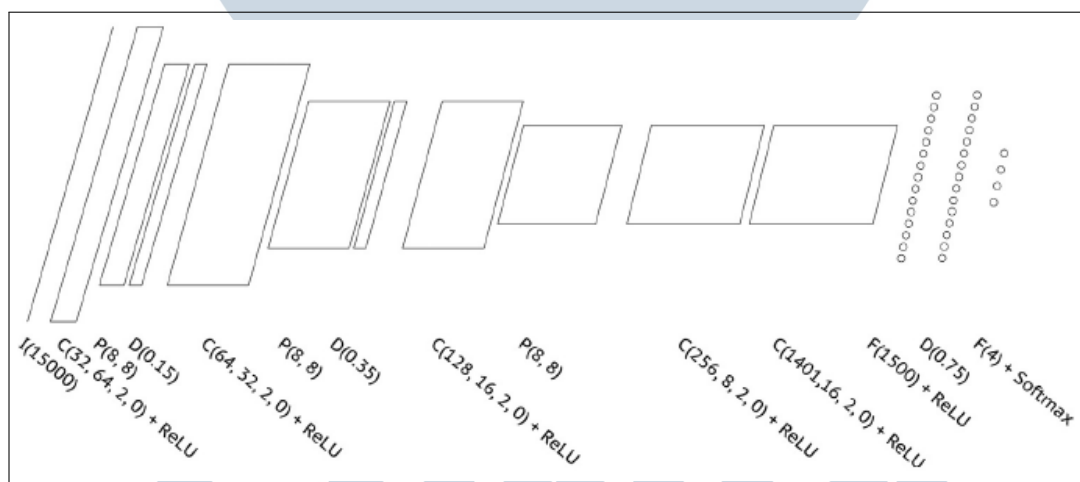


Gambar 2.9. Ilustrasi Proses CNN 2D Analisis Spektrogram

Sumber: [43]

2.8 SeismicNet (Seismic Network)

SeismicNet merupakan arsitektur *Convolutional Neural Network* (CNN) satu dimensi (1D) yang diusulkan oleh Canário *et al.* terinspirasi dari arsitektur SoundNet yang menggunakan *audio wave* atau bentuk gelombang audio mentah langsung sebagai *data input* untuk proses klasifikasi [17]. Model SeismicNet dirancang khusus untuk mengklasifikasikan kejadian (*event*) gempa vulkanik dengan memanfaatkan *raw waveform signal* secara langsung sebagai data masukan tanpa bergantung pada tahap pra-pemrosesan ekstraksi fitur manual atau transformasi sinyal menjadi domain waktu-frekuensi untuk menciptakan spektrogram [18]. Hal ini bertujuan untuk membuat tahap pembelajaran (ekstraksi fitur) hingga klasifikasi, berada dalam satu kerangka kerja yang berjalan secara *end-to-end* dengan komputasi dan waktu pemrosesan yang lebih cepat. Komponen arsitektur SeismicNet dapat dilihat pada Gambar 2.10.



Gambar 2.10. Arsitektur SeismicNet

Sumber: [18]

Adapun Rincian komponen SeismicNet dapat dilihat pada Tabel 2.4.

Tabel 2.4. Konfigurasi Arsitektur SeismicNet

Komponen	Konfigurasi
<i>Input Layer</i>	Sinyal seismik dengan panjang (<i>length</i>) tertentu dan 1 <i>channel</i> .
Lanjut ke halaman berikutnya	

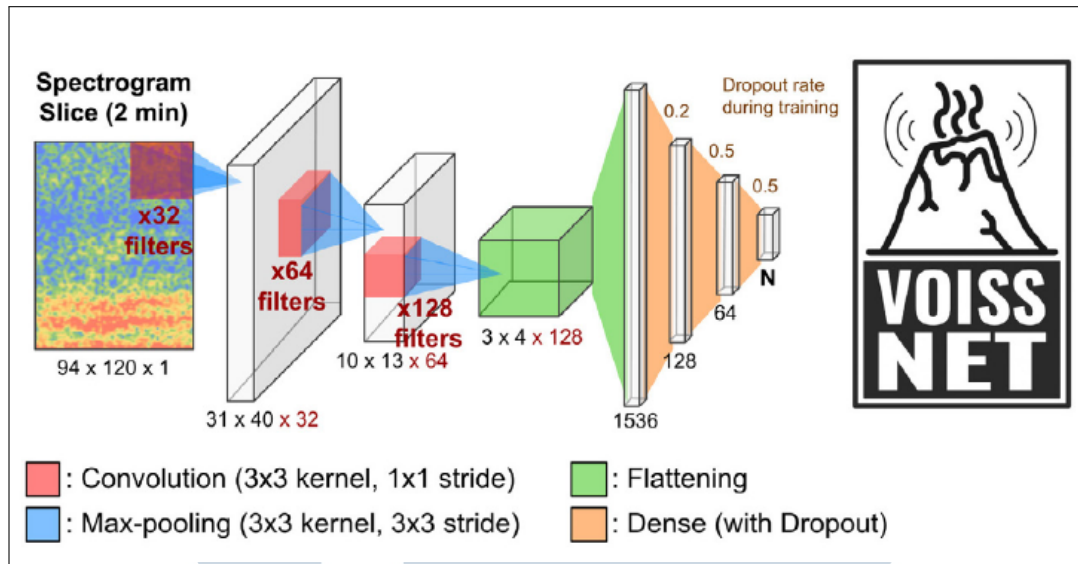
Tabel 2.4. Konfigurasi Arsitektur SeismicNet (lanjutan)

Komponen	Konfigurasi
<i>Conv1D Layer 1</i>	Filters = 32, Kernel Size = 64, Stride = 2, Padding = Valid, Activation ReLU.
<i>Max Pooling Layer 1</i>	Pool Size = 8, Stride = 8.
<i>Dropout Layer 1</i>	Dropout Rate = 0.15.
<i>Conv1D Layer 2</i>	Filters = 64, Kernel Size = 32, Stride = 2, Padding = Valid, Activation ReLU.
<i>Max Pooling Layer 2</i>	Pool Size = 8, Stride = 8.
<i>Dropout Layer 2</i>	Dropout Rate = 0.35.
<i>Conv1D Layer 3</i>	Filters = 128, Kernel Size = 16, Stride = 2, Padding = Valid, Activation ReLU.
<i>Max Pooling Layer 3</i>	Pool Size = 8, Stride = 8.
<i>Conv1D Layer 4</i>	Filters = 256, Kernel Size = 8, Stride = 2, Padding = Valid, Activation ReLU.
<i>Conv1D Layer 5</i>	Filters = 1401, Kernel Size = 16, Stride = 2, Padding = Valid, Activation ReLU.
<i>Flatten Layer</i>	Mengubah <i>feature map</i> menjadi vektor satu dimensi.
<i>Dense Layer</i>	Units = 1500, Activation ReLU.
<i>Dropout Layer</i>	Dropout Rate = 0.75.
<i>Output Layer</i>	Dense (Units = jumlah kelas), Activation Softmax.

2.9 VOISS-Net (Volcano Infrasound and Seismic Spectrogram Network)

VOISS-Net (*Volcano Infrasound and Seismic Spectrogram Network*) merupakan arsitektur CNN dua dimensi (2D) yang diusulkan oleh Tan *et al.* dirancang khusus untuk memproses data spektrogram seismik dan akustik gunung berapi [16], kemudian dikembangkan Fee *et al.* menjadi *generalized* VOISS-Net agar dapat diterapkan di berbagai gunung berapi [22]. VOISS-Net dirancang dengan tujuan utama untuk mengotomatisasi, konsisten dalam mendeteksi dan mengklasifikasikan berbagai jenis kejadian gempa melalui sinyal seismik dan infrasonik secara *real time* langsung dari data kontinu untuk berbagai gunung berapi sehingga mampu mengurangi beban kerja analisis seismik.

Komponen arsitektur VOISS-Net yang dapat dilihat pada Gambar 2.11.



Gambar 2.11. Arsitektur VOISS-Net

Sumber: [22]

Adapun rincian urutan komponen dan paramater VOISS-Net berdasarkan gambar di atas dapat dilihat pada Tabel 2.5 berikut.

Tabel 2.5. Konfigurasi Arsitektur VOISS-Net

Komponen	Konfigurasi
<i>Input Layer</i>	Spektrogram satu <i>channel</i> /kanal (<i>grayscale</i>).
<i>Conv2D Layer 1</i>	Filters = 32, Kernel Size = 3×3 , Stride = 1×1 , Padding = Same, Activation ReLU.
<i>Max Pooling Layer 1</i>	Pool Size = 3×3 , Stride = 3×3 .
<i>Conv2D Layer 2</i>	Filters = 64, Kernel Size = 3×3 , Stride = 1×1 , Padding = Same, Activation ReLU.
<i>Max Pooling Layer 2</i>	Pool Size = 3×3 , Stride = 3×3 .
<i>Conv2D Layer 3</i>	Filters = 128, Kernel Size = 3×3 , Stride = 1×1 , Padding = Same, Activation ReLU.
<i>Max Pooling Layer 3</i>	Pool Size = 3×3 , Stride = 3×3 , Dimensi keluaran = $3 \times 4 \times 128$.
Lanjut ke halaman berikutnya	

Tabel 2.5. Konfigurasi Arsitektur VOISS-Net (Lanjutan)

Komponen	Konfigurasi
<i>Flatten Layer</i>	Mengubah <i>feature map</i> menjadi vektor satu dimensi.
<i>Dropout Layer 1</i>	Dropout Rate = 0.20.
<i>Dense Layer 1</i>	Units = 128, Activation ReLU.
<i>Dropout Layer 2</i>	Dropout Rate = 0.50.
<i>Dense Layer 2</i>	Units = 64, Activation ReLU.
<i>Dropout Layer 3</i>	Dropout Rate = 0.50.
<i>Output Layer</i>	Dense (Units = jumlah kelas), Activation Softmax.

2.10 Evaluasi Performa Model

Pada penelitian ini, evaluasi performa model dilakukan melalui beberapa aspek untuk menilai kemampuan model secara menyeluruh. Evaluasi tersebut mencakup penilaian terhadap kemampuan model dalam mengklasifikasikan setiap kelas secara tepat menggunakan metrik evaluasi klasifikasi, validasi performa model menggunakan *stratified k-fold cross validation* untuk memperoleh estimasi performa yang lebih kuat terhadap variasi pembagian data, serta analisis kurva pembelajaran (*accuracy* dan *loss*) untuk mengamati dinamika proses pelatihan model.

2.10.1 Evaluasi Klasifikasi

Evaluasi klasifikasi tidak hanya berfokus pada metrik *accuracy* sebagai ukuran performa secara keseluruhan, tetapi juga pada kemampuan model dalam mengklasifikasikan setiap kelas secara tepat. Mengingat distribusi jumlah sampel antar kelas tidak seimbang (*class imbalance*), penggunaan *accuracy* berpotensi menghasilkan evaluasi yang bias terhadap kelas mayoritas. Oleh karena itu, digunakan metrik *precision*, *recall*, dan *F1-score* dengan pendekatan *macro averaging* untuk memberikan evaluasi yang setara pada setiap kelas. Selain itu, *confusion matrix* digunakan untuk menganalisis pola prediksi model, termasuk kesalahan klasifikasi antar kelas.

A Confusion Matrix

Confusion Matrix merupakan metode evaluasi yang digunakan untuk menganalisis kinerja model klasifikasi dengan membandingkan hasil prediksi terhadap label aktual. Pada klasifikasi multikelas, ukuran matriks disesuaikan dengan jumlah kelas yang digunakan. Sebagai ilustrasi, Tabel 2.6 menunjukkan struktur umum *confusion matrix* untuk empat kelas, yaitu K1, K2, K3, dan K4. Nilai pada diagonal utama, seperti n_{11} , n_{22} , n_{33} , dan n_{44} , menunjukkan jumlah prediksi yang benar pada masing-masing kelas, sedangkan nilai di luar diagonal menunjukkan kesalahan klasifikasi antar kelas. Sebagai contoh, n_{12} menunjukkan jumlah data kelas K2 yang diprediksi sebagai kelas K1.

Tabel 2.6. Ilustrasi *Confusion Matrix* Klasifikasi Multikelas

Prediksi \ Aktual	K1	K2	K3	K4
K1	n_{11}	n_{12}	n_{13}	n_{14}
K2	n_{21}	n_{22}	n_{23}	n_{24}
K3	n_{31}	n_{32}	n_{33}	n_{34}
K4	n_{41}	n_{42}	n_{43}	n_{44}

Normalized Confusion Matrix merupakan bentuk normalisasi dari *confusion matrix* yang menyajikan proporsi hasil prediksi pada setiap kelas dalam bentuk persentase [44]. Normalisasi dilakukan dengan membagi setiap nilai pada matriks terhadap total jumlah sampel pada kelas aktual (*actual class*) yang bersangkutan sehingga jumlah nilai pada setiap baris bernilai 100%. Pendekatan ini memungkinkan perbandingan performa antar kelas secara lebih adil, terutama pada dataset dengan distribusi kelas yang tidak seimbang, karena hasil evaluasi tidak dipengaruhi oleh perbedaan jumlah sampel pada masing-masing kelas.

B Accuracy

Accuracy merupakan metrik evaluasi yang digunakan untuk mengukur proporsi prediksi yang diklasifikasikan dengan benar terhadap seluruh data yang dievaluasi. Metrik ini memberikan gambaran umum mengenai kinerja model klasifikasi, namun nilainya dapat menjadi kurang akurat pada dataset dengan distribusi kelas yang tidak seimbang (*class imbalance*) karena cenderung dipengaruhi oleh kelas mayoritas sehingga tidak disarankan untuk klasifikasi

dengan *class imbalance* [44]. Secara matematis, perhitungan *accuracy* ditunjukkan pada Rumus 2.11.

$$Accuracy = \frac{\sum_{i=1}^K TP_i}{\sum_{i=1}^K (TP_i + FN_i)} \quad (2.11)$$

TP_i merupakan jumlah data pada kelas ke- i yang berhasil diprediksi dengan benar, FN_i merupakan jumlah data pada kelas ke- i yang diprediksi sebagai kelas lain, sedangkan K merupakan jumlah kelas pada proses klasifikasi.

C Macro Precision

Precision merupakan metrik evaluasi yang mengukur ketepatan model dalam memberikan prediksi positif pada setiap kelas. Nilai *macro precision* diperoleh dengan menghitung nilai *precision* untuk masing-masing kelas, kemudian mengambil rata-ratanya sehingga setiap kelas memiliki bobot yang sama dalam proses evaluasi [44]. Pendekatan ini sesuai digunakan pada dataset dengan distribusi kelas yang tidak seimbang (*class imbalance*) karena setiap kelas memberikan kontribusi yang sama terhadap nilai evaluasi akhir. Secara matematis, perhitungan *Macro Precision* ditunjukkan pada Rumus 2.12.

$$Macro\ Precision = \frac{1}{K} \sum_{i=1}^K \frac{TP_i}{TP_i + FP_i} \quad (2.12)$$

TP_i merupakan jumlah data pada kelas ke- i yang berhasil diprediksi dengan benar, FP_i merupakan jumlah data dari kelas lain yang salah diprediksi sebagai kelas ke- i , sedangkan K merupakan jumlah kelas pada proses klasifikasi.

D Macro Recall

Recall merupakan metrik evaluasi yang mengukur kemampuan model dalam mengenali data aktual pada setiap kelas. Nilai *macro recall* diperoleh dengan menghitung nilai *recall* untuk masing-masing kelas, kemudian mengambil rata-ratanya sehingga setiap kelas memiliki bobot yang sama dalam proses evaluasi [44]. Pendekatan ini sesuai digunakan pada dataset dengan distribusi kelas yang tidak seimbang (*class imbalance*) karena setiap kelas memberikan kontribusi yang sama terhadap nilai evaluasi akhir. Secara matematis, perhitungan *Macro Recall*

ditunjukkan pada Rumus 2.13.

$$Macro Recall = \frac{1}{K} \sum_{i=1}^K \frac{TP_i}{TP_i + FN_i} \quad (2.13)$$

TP_i merupakan jumlah data pada kelas ke- i yang berhasil diprediksi benar, FN_i merupakan jumlah data aktual pada kelas ke- i yang salah diprediksi, sedangkan K merupakan jumlah kelas yang digunakan dalam proses klasifikasi.

E Macro F1-Score

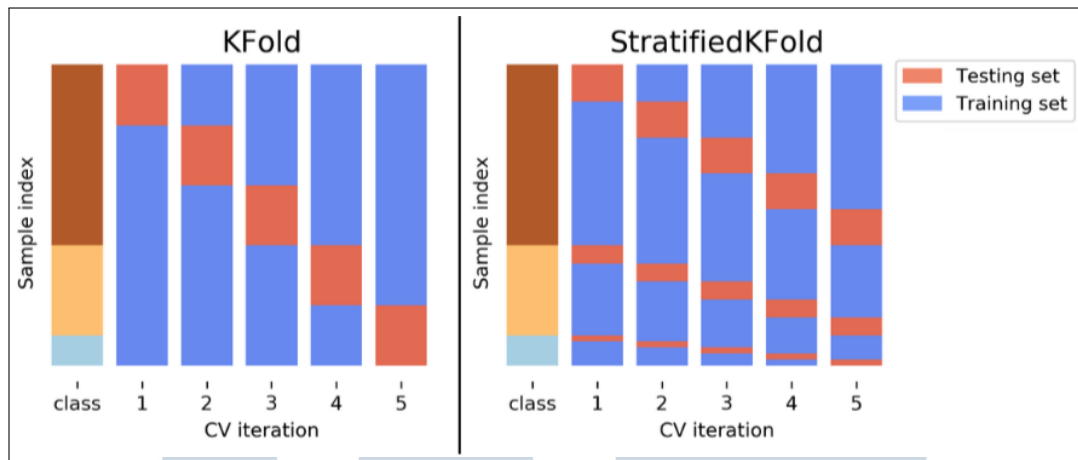
F1-Score merupakan metrik evaluasi yang menggabungkan nilai *precision* dan *recall* menggunakan rata-rata harmonik (*harmonic mean*) sehingga menghasilkan ukuran performa yang lebih seimbang pada setiap kelas [44]. Nilai *macro F1 score* diperoleh dengan menghitung *F1-Score* untuk masing-masing kelas, kemudian mengambil rata-ratanya sehingga setiap kelas memiliki bobot yang sama dalam proses evaluasi. Pendekatan ini sesuai digunakan pada dataset dengan distribusi kelas yang tidak seimbang (*class imbalance*) karena setiap kelas memberikan kontribusi yang sama terhadap nilai evaluasi akhir. Secara matematis, perhitungan *Macro F1-Score* ditunjukkan pada Rumus 2.14.

$$Macro F1 = \frac{1}{K} \sum_{i=1}^K \left(2 \times \frac{Precision_i \times Recall_i}{Precision_i + Recall_i} \right) \quad (2.14)$$

Pada Rumus 2.14, $Precision_i$ merupakan nilai *precision* pada kelas ke- i , $Recall_i$ merupakan nilai *recall* pada kelas ke- i , sedangkan K merupakan jumlah kelas yang digunakan dalam proses klasifikasi.

2.10.2 Stratified K-Fold Cross Validation

Cross validation merupakan teknik evaluasi yang digunakan untuk mengukur kemampuan generalisasi model dengan membagi dataset menjadi beberapa bagian yang digunakan secara bergantian sebagai data pelatihan dan data pengujian. Melalui pendekatan ini, model dievaluasi pada beberapa kombinasi data pelatihan dan data pengujian sehingga estimasi performa yang diperoleh menjadi lebih akurat [25].



Gambar 2.12. Ilustrasi *Stratified K-Fold Cross Validation*

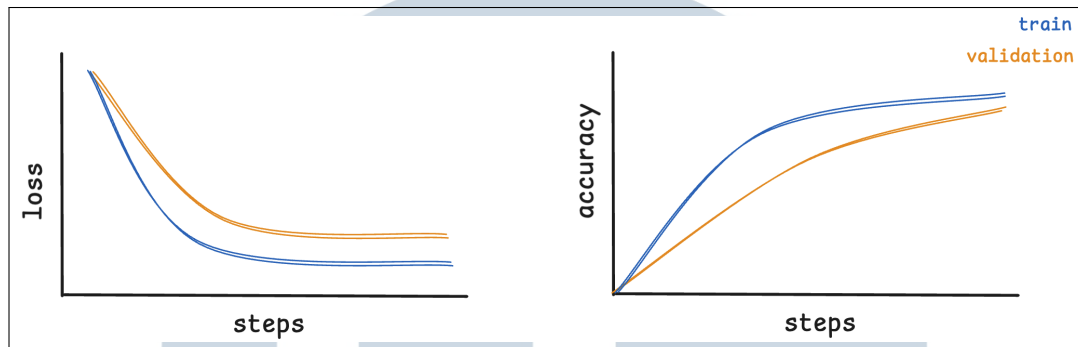
Sumber: [45]

Salah satu teknik *cross-validation* yang umum digunakan adalah *K-Fold Cross Validation* seperti diilustrasikan pada Gambar 2.12. Pada metode ini, satu *fold* digunakan sebagai data pengujian, sedangkan $k - 1$ *fold* lainnya digunakan sebagai data pelatihan. Proses tersebut diulang sebanyak k iterasi hingga setiap *fold* digunakan satu kali sebagai data pengujian [25]. Namun, pada dataset dengan distribusi kelas yang tidak seimbang (*class imbalance*), pembagian data dapat menghasilkan proporsi kelas yang berbeda pada setiap *fold*, sehingga kelas minoritas berpotensi memiliki jumlah sampel yang sangat sedikit atau bahkan tidak tersedia pada *fold* tertentu. Kondisi tersebut dapat memengaruhi hasil evaluasi model [25]. Oleh karena itu, digunakan *Stratified K-Fold Cross Validation* yang mempertahankan proporsi setiap kelas pada seluruh *fold* agar tetap mendekati distribusi kelas pada dataset asli, sehingga distribusi kelas pada setiap *fold* tetap mendekati distribusi kelas pada dataset asli dan menghasilkan evaluasi model yang lebih representatif [45].

2.10.3 Learning Curves (Accuracy & Loss)

Learning curves atau kurva pembelajaran merupakan visualisasi yang menunjukkan perubahan performa model selama proses pelatihan pada setiap *epoch*. Kurva ini digunakan untuk memantau perkembangan proses pelatihan serta mengevaluasi kemampuan generalisasi model melalui perbandingan performa pada data pelatihan (*training*) dan data validasi (*validation*), sehingga kondisi overfitting maupun underfitting dapat diidentifikasi [46]. Umumnya, *learning*

curves divisualisasikan menggunakan kurva *loss* dan *accuracy*, seperti ditunjukkan pada Gambar 2.13.



Gambar 2.13. Kurva Pembelajaran - *Loss* dan *Accuracy*

Sumber: [46]

Kurva *loss* menunjukkan perubahan nilai kesalahan model selama proses pelatihan, sedangkan kurva *accuracy* menunjukkan perubahan tingkat ketepatan prediksi model pada setiap *epoch*. Secara umum, proses pelatihan yang baik ditunjukkan oleh nilai *loss* yang cenderung menurun dan nilai *accuracy* yang cenderung meningkat hingga keduanya mencapai kondisi yang relatif stabil. Selain itu, selisih antara kurva *training* dan *validation* yang relatif kecil menunjukkan kemampuan generalisasi model yang baik. Sebaliknya, selisih yang besar dapat mengindikasikan overfitting, sedangkan performa yang masih rendah pada kedua kurva dapat mengindikasikan underfitting [46].

2.11 Evaluasi Kompleksitas Komputasi

Evaluasi kompleksitas komputasi dilakukan untuk mengukur kebutuhan sumber daya model, baik dari aspek penyimpanan maupun komputasi. Evaluasi ini dilakukan berdasarkan jumlah parameter, ukuran model (*model size*), dan *Floating Point Operations* (FLOPs). Hasil evaluasi digunakan untuk menilai efisiensi model, terutama dalam mendukung proses *deployment*. Secara umum, model dengan kompleksitas yang lebih tinggi memerlukan sumber daya komputasi dan memori yang lebih besar, sehingga perlu dipertimbangkan keseimbangan antara performa klasifikasi dan efisiensi komputasi.

2.11.1 Jumlah Parameter dan Ukuran Model

Jumlah parameter model menggambarkan kompleksitas arsitektur berdasarkan total parameter yang dapat dipelajari pada seluruh lapisan jaringan, semakin besar jumlah parameter maka semakin tinggi kebutuhan memori dan kompleksitas model yang dihasilkan [47]. Sementara itu, ukuran model (*model size*) menunjukkan besaran ruang penyimpanan yang diperlukan untuk menyimpan model hasil pelatihan sehingga dapat digunakan pada proses implementasi [48]. Dalam implementasinya, jumlah parameter dapat diperoleh melalui fungsi `model.summary()`, sedangkan ukuran model dapat diketahui berdasarkan ukuran file model yang disimpan dalam format `.keras` [48].

2.11.2 Floating Point Operations (FLOPs)

Floating Point Operations (FLOPs) merupakan metrik yang digunakan untuk mengukur jumlah operasi aritmetika *floating-point* yang dilakukan model selama proses inferensi [49]. Nilai FLOPs memberikan gambaran mengenai kompleksitas komputasi model, semakin besar nilai FLOPs maka semakin tinggi kebutuhan komputasi untuk memproses data. Oleh karena itu, FLOPs banyak digunakan untuk mengevaluasi efisiensi komputasi model, terutama pada implementasi model *deep learning* dengan keterbatasan sumber daya komputasi [48, 49]. Nilai FLOPs dapat diperoleh menggunakan TensorFlow Profiler yang menghitung jumlah operasi *floating-point* yang dilakukan model selama satu kali proses *forward pass*, sehingga FLOPs dapat digunakan sebagai indikator untuk membandingkan kompleksitas komputasi antar model *deep learning*.

2.12 Evaluasi Waktu Komputasi

Evaluasi waktu komputasi dilakukan untuk mengukur efisiensi model dari sisi waktu pemrosesan selama proses pelatihan maupun inferensi. Evaluasi ini mencakup total durasi waktu pelatihan dan waktu inferensi guna memberikan gambaran mengenai kebutuhan waktu model dalam mempelajari data serta menghasilkan prediksi terhadap data masukan.

2.12.1 Total Waktu Pelatihan

Total waktu pelatihan digunakan untuk mengukur durasi waktu yang dibutuhkan selama proses pelatihan model pada seluruh *iterasi* dalam *Stratified K-Fold Cross Validation* dalam satuan detik atau menit. Pengukuran dilakukan dengan menghitung selisih waktu antara awal pelatihan pada *fold* pertama dan akhir pelatihan pada *fold* terakhir, seperti ditunjukkan pada Rumus 2.15.

$$T_{train} = t_{end} - t_{start} \quad (2.15)$$

T_{train} adalah total waktu pelatihan seluruh *fold*, t_{start} adalah waktu saat pelatihan pada *fold* pertama dimulai, t_{end} adalah waktu saat pelatihan pada *fold* terakhir selesai.

2.12.2 Waktu Inferensi

Waktu inferensi digunakan untuk mengukur waktu yang dibutuhkan model dalam menghasilkan prediksi terhadap data masukan. Pengukuran dilakukan menggunakan model terbaik hasil pelatihan dengan menghitung waktu prediksi setiap sampel secara individual untuk memperoleh rata-rata waktu inferensi per sampel (*inference time latency*) yang dinyatakan dalam satuan *milliseconds*. Sebelum pengukuran dilakukan, proses *warm-up* diterapkan untuk mengurangi pengaruh inisialisasi awal model terhadap hasil pengukuran. Perhitungan rata-rata waktu inferensi dari seluruh sampel yang diuji, dinyatakan dengan Rumus 2.16.

$$\bar{T}_{inf} = \frac{1}{N} \sum_{i=1}^N T_{inf}^{(i)} \quad (2.16)$$

$\bar{T}_{inf}$ merupakan rata-rata waktu inferensi per sampel, $T_{inf}^{(i)}$ merupakan waktu inferensi pada sampel ke- i , sedangkan N merupakan jumlah sampel data yang digunakan dalam proses pengukuran waktu inferensi.