BAB 2
LANDASAN TEORI

Pada bagian ini dijabarkan teori-teori yang mendasari penelitian secara menyeluruh, dimulai dari kajian terhadap penelitian terdahulu hingga konsep-konsep teknis yang digunakan dalam perancangan model. Pembahasan mencakup studi literatur mengenai pendekatan deteksi *deepfake*, konsep dasar *deepfake* beserta jejak artefaknya, kategori pendekatan deteksi berbasis *deep learning*, serta komponen-komponen utama yang digunakan, yaitu EfficientNet, *Convolutional Block Attention Module* (CBAM), dan dekomposisi frekuensi berbasis *Discrete Cosine Transform* (DCT). Bagian akhir menguraikan dataset dan metrik evaluasi yang menjadi dasar pengukuran kinerja model.

2.1 Studi Literatur

Sejumlah penelitian terdahulu telah membahas peningkatan deteksi manipulasi wajah berbasis *deepfake* melalui berbagai pendekatan, mulai dari analisis pada domain spasial, pemanfaatan domain frekuensi, integrasi kedua domain melalui arsitektur *dual-branch*, hingga pengujian kemampuan generalisasi secara *cross-dataset*. Bagian ini meninjau penelitian-penelitian tersebut secara berurutan untuk menempatkan posisi penelitian yang dilakukan.

Pendekatan berbasis domain spasial menjadi titik awal sebagian besar metode deteksi *deepfake*. Yadav dan Mangalampalli [17] mengembangkan arsitektur hibrida yang menggabungkan CNN, *Bidirectional LSTM*, dan *transformer encoder* untuk menangkap fitur spasial sekaligus dinamika temporal pada video. Dengan MobileNetV2 sebagai *backbone* dan pengujian pada FaceForensics++ serta DFDC, model tersebut mencapai F1 sebesar 90,6% dan AUC 98,5%, serta menunjukkan ketahanan terhadap penurunan kualitas video. Studi ini menegaskan bahwa CNN efektif menangkap artefak spasial pada tingkat *frame*, tetapi analisis spasial secara terpisah cenderung kurang memadai pada kondisi nyata yang melibatkan kompresi maupun ketidakstabilan *frame*. Pada arah yang sama, Tobing dkk. [16] menerapkan CNN untuk membedakan citra wajah asli dan *deepfake* serta memperoleh akurasi 81,3%, namun melaporkan gejala *overfitting* berupa kenaikan *validation loss* seiring bertambahnya *epoch* dan menegaskan perlunya variasi dataset yang lebih luas agar model mampu menggeneralisasi. Temuan ini

memperlihatkan bahwa pendekatan spasial murni memang rentan *overfitting* dan terbatas kemampuannya ketika dihadapkan pada data di luar distribusi pelatihan.

Keterbatasan tersebut mendorong pemanfaatan informasi pada domain frekuensi sebagai pelengkap. Gong et al. [18] mengusulkan kerangka deteksi yang memadukan fitur spasial dan frekuensi dengan EfficientNet sebagai *backbone*, serta memanfaatkan *Discrete Cosine Transform* (DCT) untuk mengekstraksi jejak manipulasi pada domain frekuensi. Kerangka tersebut melibatkan modul *cross-attention fusion*, *guided attention*, dan *multi-scale feature fusion* untuk memperkuat interaksi antar-domain. Metode ini menunjukkan ketahanan terhadap citra terkompresi, dengan AUC sebesar 89,95% pada subset FF++ berkualitas rendah dan peningkatan AUC rata-rata 3,9% dibandingkan metode pembanding. Temuan ini memperlihatkan bahwa informasi frekuensi mampu menonjolkan artefak yang tidak terlihat pada domain warna, khususnya ketika artefak spasial melemah akibat kompresi.

Menyadari bahwa kedua domain bersifat saling melengkapi, sejumlah penelitian mengembangkan arsitektur *dual-branch* yang memproses fitur spasial dan frekuensi secara bersamaan. Wang et al. [19] mengusulkan jaringan fusi lintas-domain yang memadukan fitur warna pada domain spasial dengan fitur *wavelet* dan fitur residual pada domain frekuensi, lalu menyatukannya melalui modul fusi adaptif berbasis *gated convolution* sehingga kontribusi tiap domain dapat diatur secara dinamis. Dengan ResNet-34 sebagai *backbone*, metode tersebut mencapai akurasi di atas 98% pada FF++. Sejalan dengan itu, Zhai et al. [20] merancang jaringan fusi spasial–frekuensi dua aliran yang memanfaatkan *spatial-channel attention* pada cabang spasial dan *focused linear attention* pada cabang frekuensi, lalu menggabungkan keduanya melalui modul fusi khusus. Kedua penelitian ini menunjukkan bahwa integrasi dua domain memberikan representasi yang lebih komprehensif, tetapi efektivitasnya sangat bergantung pada cara kedua representasi diseleksi dan disatukan sehingga penggabungan yang sederhana belum tentu memberikan peningkatan yang berarti.

Selain rancangan arsitektur, kemampuan generalisasi menjadi perhatian utama karena performa pada satu dataset tidak menjamin keandalan pada dataset lain. Pengujian *cross-dataset* telah menjadi praktik umum untuk menilai aspek ini. Zhai et al. [20] mengevaluasi metodenya tidak hanya pada FF++, tetapi juga pada Celeb-DF, DFD, dan DFDC, dan melaporkan bahwa integrasi fitur frekuensi dengan spasial meningkatkan kemampuan generalisasi. Gong et al. [18] juga melakukan pengujian serupa dengan melatih model pada FF++ lalu mengujinya pada Celeb-DF,

DFDC, dan WildDeepfake, dan mencatat bahwa performa seluruh metode menurun pada skenario lintas dataset. Petmezas [21] secara khusus mengevaluasi arsitektur berbasis Transformer dan CNN pada skenario *cross-dataset* dan memperoleh capaian terbaik pada kisaran AUC 0,801, yang menegaskan bahwa generalisasi tetap menjadi tantangan terbuka. Tinjauan sistematis oleh Ramanaharan et al. [22] memperkuat hal ini dengan menyimpulkan bahwa generalisasi terhadap dataset yang tidak dikenal masih menjadi persoalan yang belum terselesaikan, sementara survei oleh Qureshi et al. [23] juga menempatkan keterbatasan generalisasi sebagai salah satu isu utama dalam deteksi *deepfake*.

Berdasarkan tinjauan tersebut, integrasi fitur spasial dan frekuensi melalui arsitektur *dual-branch* serta pemanfaatan mekanisme atensi merupakan arah yang konsisten dipilih dalam penelitian terdahulu, dan pengujian *cross-dataset* telah menjadi tolok ukur yang lazim. Namun, sebagian besar penelitian memusatkan perhatian pada perancangan modul fusi atau pencapaian performa secara keseluruhan, sementara kajian yang secara khusus mengisolasi kontribusi penempatan mekanisme atensi pada masing-masing cabang dalam kerangka *dual-branch* masih terbatas. Hal inilah yang menjadi fokus penelitian ini, yaitu mengkaji integrasi fitur spasial dan frekuensi yang diseleksi secara adaptif pada tiap cabang dan mengevaluasi pengaruhnya terhadap kemampuan generalisasi melalui pengujian *cross-dataset*.

2.2 Deepfake

Deepfake adalah konten media sintetis, umumnya video atau gambar, yang dihasilkan melalui teknik *deep learning*, dengan wajah seseorang digantikan atau dimanipulasi sedemikian rupa sehingga hasilnya sulit dibedakan dari konten asli oleh mata manusia. Istilah ini berasal dari kombinasi kata *deep learning* dan *fake* [24]. Teknik generasinya meliputi beberapa paradigma utama.

Face Swapping adalah teknik yang menggantikan identitas wajah seseorang dalam video dengan identitas orang lain. Metode ini menggunakan *autoencoder* berpasangan atau *Generative Adversarial Network* (GAN) untuk memetakan fitur wajah sumber ke wajah target. DeepFakes dan FaceSwap dalam dataset FaceForensics++ merupakan representasi metode ini [25].

Face Reenactment memanipulasi ekspresi dan gerakan wajah seseorang, termasuk posisi kepala, ekspresi wajah, dan gerakan mulut, tanpa mengganti identitasnya. Face2Face dan NeuralTextures adalah contoh metode *face*

reenactment yang memanfaatkan model 3D *morphable* dan *neural rendering* [25].

Dari perspektif forensik, proses manipulasi *deepfake* meninggalkan jejak artefak yang dapat dieksploitasi untuk deteksi. Pada domain spasial, artefak ini dapat berupa inkonsistensi tekstur di batas wajah, *blending artifacts*, dan pola kompresi yang tidak konsisten. Pada domain frekuensi, proses *upsampling* dalam *pipeline* GAN meninggalkan pola periodik yang khas pada frekuensi tinggi [26, 24].

2.3 Deteksi Deepfake Berbasis *Deep Learning*

Pendekatan berbasis *deep learning* mendominasi penelitian deteksi *deepfake* karena kemampuannya mempelajari representasi fitur secara langsung dari data. Secara umum, metode yang ada dapat dikategorikan berdasarkan domain analisisnya menjadi tiga kelompok, yaitu pendekatan berbasis domain spasial, domain frekuensi, dan gabungan keduanya.

2.3.1 Pendekatan Berbasis Domain Spasial

Pendekatan berbasis domain spasial memanfaatkan CNN untuk mengekstrak fitur piksel dan pola kesalahan (*error pattern*) dari citra wajah. Arsitektur seperti Xception mengandalkan *depthwise separable convolution* untuk menangkap pola visual artefak manipulasi. Meskipun efektif dalam skenario *within-domain*, pendekatan ini rentan terhadap perubahan distribusi data (*domain shift*) karena terlalu bergantung pada artefak visual yang spesifik terhadap metode manipulasi tertentu [24].

2.3.2 Pendekatan Berbasis Domain Frekuensi

Pendekatan berbasis domain frekuensi mengeksploitasi perbedaan spektral antara citra asli dan citra hasil manipulasi. Ketika sebuah wajah palsu dihasilkan oleh GAN, proses *upsampling* menghasilkan spektrum frekuensi yang berbeda dari wajah asli, khususnya pada komponen frekuensi tinggi. Perbedaan ini dimanfaatkan sebagai sinyal pembeda yang seringkali tidak terlihat pada domain spasial [26]. Namun, pendekatan frekuensi murni memiliki keterbatasan karena tidak memanfaatkan konteks spasial yang kaya dari domain citra [27].

2.3.3 Pendekatan *Dual-Branch* Spasial-Frekuensi

Guna menggabungkan keunggulan kedua domain, pendekatan *dual-branch* memproses informasi spasial dan frekuensi secara paralel, kemudian menggabungkan representasinya melalui mekanisme fusi. Eksperimen lintas penelitian secara konsisten menunjukkan bahwa arsitektur *dual-branch* menghasilkan generalisasi yang lebih baik dibandingkan pendekatan berbasis domain tunggal, khususnya dalam evaluasi *cross-dataset* [27, 28].

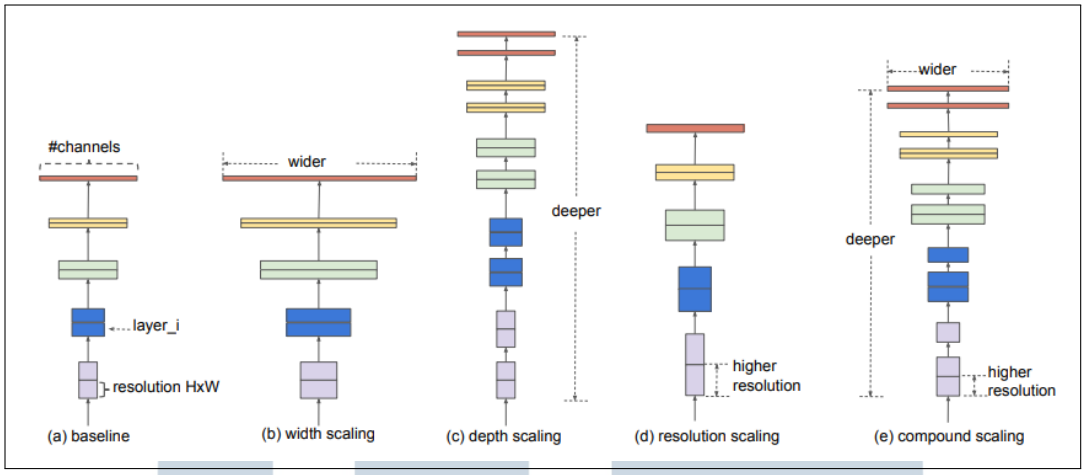
2.4 EfficientNet

EfficientNet adalah keluarga arsitektur *Convolutional Neural Network* (CNN) yang diperkenalkan oleh Tan dan Le [29] pada tahun 2019 melalui konferensi ICML. Kontribusi utamanya adalah metode penskalaan *compound scaling*, yang dilatari oleh pengamatan bahwa metode penskalaan konvensional hanya menskalakan satu dimensi jaringan secara terpisah, baik kedalaman (*depth*), lebar (*width*), maupun resolusi (*resolution*), sehingga peningkatan akurasi yang diperoleh cepat mengalami saturasi. Alih-alih menskalakan ketiga dimensi tersebut secara arbitrer, *compound scaling* menyeimbangkan ketiganya secara serentak menggunakan satu koefisien tunggal ϕ yang ditentukan oleh pengguna sesuai dengan jumlah *resource* komputasi yang tersedia. Secara formal, penskalaan dilakukan sebagaimana ditunjukkan pada Persamaan 2.1.

$$\text{depth} : d = \alpha^{\phi}, \quad \text{width} : w = \beta^{\phi}, \quad \text{resolution} : r = \gamma^{\phi} \tag{2.1}$$

α, β, γ adalah konstanta yang ditentukan melalui *grid search* berskala kecil pada jaringan *baseline*, dengan batasan $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$ serta $\alpha \geq 1, \beta \geq 1, \gamma \geq 1$ [29]. Batasan ini memastikan bahwa setiap kenaikan satu satuan ϕ secara konsisten meningkatkan kebutuhan *floating point operations* (FLOPS) total kurang lebih sebesar 2^{ϕ}.

Gambar 2.1 mengilustrasikan perbedaan *compound scaling* dengan metode penskalaan konvensional. Jaringan *baseline* (a) hanya diskalakan pada satu dimensi saja oleh metode konvensional, baik lebar (b), kedalaman (c), maupun resolusi (d), sedangkan *compound scaling* (e) menambah ketiga dimensi tersebut secara bersamaan dengan rasio tetap.



Gambar 2.1. Ilustrasi model scaling: baseline, penskalaan satu dimensi (lebar, kedalaman, resolusi), dan compound scaling

Sumber: [29]

Baseline arsitekturnya, EfficientNet-B0, diperoleh melalui *neural architecture search* (NAS) yang mengoptimalkan akurasi dan FLOPS secara bersamaan, dengan blok utama berupa *Mobile Inverted Bottleneck Convolution* (MBConv) yang menggunakan *depthwise separable convolution* dan modul *squeeze-and-excitation* untuk efisiensi parameter [29]. Koefisien α, β, γ kemudian dicari sekali pada EfficientNet-B0 ini (dengan $\phi = 1$), lalu digunakan tetap untuk menurunkan varian EfficientNet-B1 hingga B7 dengan menaikkan nilai ϕ. Arsitektur lengkap EfficientNet-B0 yang terdiri atas sembilan *stage* ditunjukkan pada Tabel 2.1.

Tabel 2.1. Arsitektur baseline EfficientNet-B0

Stage	Operator	Resolusi	#Channel	#Layer
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	28×28	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

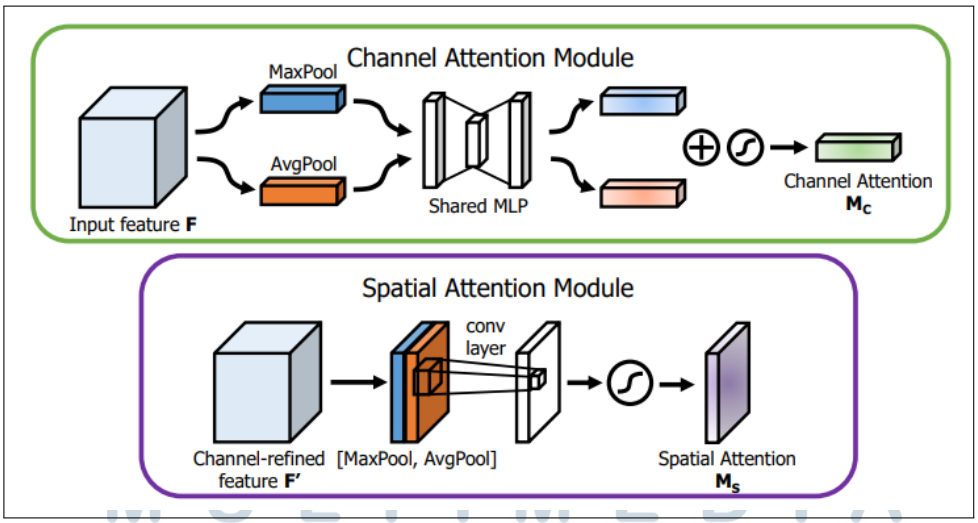
Sumber: [29]

2.5 Convolutional Block Attention Module (CBAM)

CBAM adalah modul atensi untuk CNN yang diperkenalkan oleh Woo et al. [30] pada ECCV 2018. Modul ini bertujuan memperkaya representasi fitur dengan cara menonjolkan fitur yang relevan dan menekan fitur yang tidak relevan, melalui dua submodul yang diterapkan secara berurutan (*sequential*), yaitu *Channel Attention Module* (CAM) dan *Spatial Attention Module* (SAM). Diberikan peta fitur antara $\mathbf{F} \in \mathbb{R}^{C \times H \times W}$ sebagai input, CBAM secara berurutan menghasilkan peta atensi *channel* $\mathbf{M}_c \in \mathbb{R}^{C \times 1 \times 1}$ dan peta atensi spasial $\mathbf{M}_s \in \mathbb{R}^{1 \times H \times W}$. Keseluruhan proses atensi ini diformulasikan pada Persamaan 2.2.

$$\mathbf{F}' = \mathbf{M}_c(\mathbf{F}) \otimes \mathbf{F}, \qquad \mathbf{F}'' = \mathbf{M}_s(\mathbf{F}') \otimes \mathbf{F}' \tag{2.2}$$

dengan $\otimes$ menyatakan perkalian elemen per elemen (*element-wise multiplication*). Pada saat perkalian, nilai atensi disebarkan (*broadcast*) sesuai dimensi yang sesuai: nilai atensi *channel* disebarkan sepanjang dimensi spasial, dan nilai atensi spasial disebarkan sepanjang dimensi *channel*. $\mathbf{F}''$ merupakan keluaran fitur akhir yang telah disempurnakan (*refined*). Gambar 2.2 menunjukkan diagram proses komputasi dari masing-masing submodul atensi tersebut.



Gambar 2.2. Diagram setiap submodul atensi pada CBAM: channel attention module dan spatial attention module

Sumber: [30]

Channel Attention bertujuan memilih *channel* fitur yang paling relevan, karena setiap *channel* dari peta fitur dapat dipandang sebagai sebuah pendeteksi fitur

(*feature detector*) yang merepresentasikan "apa" (*what*) yang bermakna pada citra input. Untuk menghitungnya, dimensi spasial dari peta fitur input terlebih dahulu diringkas (*squeeze*) menggunakan operasi *average-pooling* dan *max-pooling* secara bersamaan, menghasilkan dua deskriptor konteks spasial $\mathbf{F}^c_{avg}$ dan $\mathbf{F}^c_{max}$. Kedua deskriptor tersebut kemudian diteruskan ke sebuah jaringan *multi-layer perceptron* (MLP) berbagi bobot dengan satu *hidden layer*, di mana ukuran aktivasi *hidden layer* tersebut direduksi menjadi $\mathbb{R}^{C/r\times 1\times 1}$ untuk mengurangi *overhead* parameter, dengan r adalah rasio reduksi (*reduction ratio*). Keluaran dari jaringan tersebut kemudian digabungkan menggunakan penjumlahan elemen per elemen. Secara formal, atensi *channel* dihitung sebagaimana ditunjukkan pada Persamaan 2.3.

$$\mathbf{M}_c(\mathbf{F}) = \sigma(\text{MLP}(\text{AvgPool}(\mathbf{F})) + \text{MLP}(\text{MaxPool}(\mathbf{F}))) \tag{2.3}$$

dengan σ menyatakan fungsi sigmoid, $\mathbf{W}_0 \in \mathbb{R}^{C/r\times C}$ dan $\mathbf{W}_1 \in \mathbb{R}^{C\times C/r}$ sebagai bobot MLP yang dipakai bersama untuk kedua deskriptor, serta fungsi aktivasi ReLU yang mengikuti $\mathbf{W}_0$.

Spatial Attention bertujuan mengidentifikasi lokasi spasial yang paling informatif, yang bersifat komplementer terhadap atensi *channel* karena merepresentasikan "di mana" (*where*) letak informasi yang penting. Untuk menghitungnya, dilakukan agregasi informasi *channel* dari peta fitur yang telah disempurnakan oleh atensi *channel* ($\mathbf{F}'$) menggunakan operasi *average-pooling* dan *max-pooling* di sepanjang sumbu *channel*, menghasilkan dua deskriptor 2D yaitu $\mathbf{F}^s_{avg} \in \mathbb{R}^{1\times H\times W}$ dan $\mathbf{F}^s_{max} \in \mathbb{R}^{1\times H\times W}$. Kedua deskriptor tersebut kemudian digabungkan (*concatenate*) dan dilewatkan ke sebuah *layer* konvolusi untuk menghasilkan peta atensi spasial, sebagaimana ditunjukkan pada Persamaan 2.4.

$$\mathbf{M}_s(\mathbf{F}') = \sigma\left(f^{7\times 7}\left([\text{AvgPool}(\mathbf{F}');\text{MaxPool}(\mathbf{F}')]\right)\right) \tag{2.4}$$

dengan $f^{7\times 7}$ menyatakan operasi konvolusi dengan ukuran *filter* 7×7 dan $[\cdot;\cdot]$ menyatakan operasi penggabungan sepanjang dimensi *channel*.

Kedua submodul atensi tersebut dapat disusun secara paralel maupun berurutan. Woo et al. [30] menunjukkan bahwa penyusunan secara berurutan memberikan hasil yang lebih baik dibandingkan penyusunan secara paralel, dan di antara dua kemungkinan urutan berurutan, urutan *channel* diikuti spasial (*channel-first*) sedikit lebih baik dibandingkan urutan sebaliknya. Atas dasar tersebut, CBAM

secara final dirancang dengan menerapkan CAM terlebih dahulu, kemudian diikuti oleh SAM, sebagaimana telah diformulasikan pada Persamaan 2.2.

Keluaran dari rangkaian kedua submodul tersebut, $\mathbf{F}''$, bukan berupa skor klasifikasi maupun vektor fitur baru, melainkan peta fitur yang telah disempurnakan (*refined feature map*) dengan dimensi yang identik terhadap masukannya, yaitu tetap $\mathbb{R}^{C\times H\times W}$. Karena peta atensi $\mathbf{M}_c$ dan $\mathbf{M}_s$ bernilai pada rentang $[0,1]$ akibat fungsi sigmoid, perkalian elemen per elemen hanya menyesuaikan bobot penekanan pada tiap *channel* dan lokasi spasial: bagian yang informatif dipertahankan mendekati nilai aslinya (bobot mendekati 1), sedangkan bagian yang tidak relevan diredam mendekati nol. Dengan kata lain, CBAM tidak menambah atau mengurangi dimensi fitur, melainkan mengalibrasi ulang respons *backbone*. Sifat preservasi dimensi ini, dipadukan dengan rancangannya yang ringan (*lightweight*), memungkinkan CBAM disisipkan langsung di antara blok konvolusi mana pun pada berbagai arsitektur CNN dan dilatih secara *end-to-end* bersama jaringan dasarnya.

2.6 Discrete Cosine Transform (DCT)

Discrete Cosine Transform (DCT) adalah metode transformasi sinyal yang menyatakan suatu sinyal sebagai penjumlahan dari sejumlah fungsi kosinus dengan frekuensi yang berbeda-beda sebagai fungsi basisnya [31]. DCT memindahkan representasi sinyal dari domain spasial ke domain frekuensi dengan redundansi informasi yang minimal, karena fungsi-fungsi basis kosinus yang digunakannya membentuk basis ortogonal. Keunggulan utama DCT adalah kemampuan pemadatan energi (*energy compaction*) yang tinggi, sehingga sebagian besar informasi sinyal terkonsentrasi hanya pada beberapa koefisien frekuensi rendah saja. Sifat inilah yang menjadikan DCT sebagai transformasi dasar pada standar kompresi citra yang umum digunakan seperti JPEG, serta pada standar kompresi video seperti MPEG dan H.264 [31].

Terdapat beberapa varian DCT (DCT-I hingga DCT-VIII), namun varian yang paling umum digunakan pada pemrosesan dan kompresi citra adalah DCT-II, karena memiliki sifat pemadatan energi yang kuat [31]. Untuk sebuah citra berukuran $N \times N$ dengan fungsi intensitas $f(x,y)$, koefisien DCT-II dua dimensi pada orde (p,q) dihitung sebagaimana ditunjukkan pada Persamaan 2.5.

$$C_{pq} = \rho(p)\,\rho(q) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)\, D_p(x)\, D_q(y) \tag{2.5}$$

dengan $0 \leq p,q,x,y \leq N-1$. Fungsi kernel kosinus dari basis ortogonal $D_n(t)$ didefinisikan pada Persamaan 2.6.

$$D_n(t) = \cos\left(\frac{(2t+1)\,n\,\pi}{2N}\right) \tag{2.6}$$

sedangkan $\rho(n)$ merupakan faktor normalisasi yang didefinisikan pada Persamaan 2.7.

$$\rho(n) = \begin{cases} \sqrt{1/N} & \text{jika } n = 0 \\ \sqrt{2/N} & \text{selainnya} \end{cases} \tag{2.7}$$

Salah satu sifat penting DCT pada citra adalah tata letak distribusi frekuensinya yang teratur: koefisien frekuensi rendah terletak pada sudut kiri atas spektrum, sedangkan koefisien frekuensi tinggi terletak pada sudut kanan bawah. Komponen frekuensi rendah merepresentasikan informasi global atau gambaran kasar dari citra, sementara komponen frekuensi tinggi merepresentasikan informasi detail berskala kecil seperti tepi (*edge*) dan tekstur. Tata letak yang teratur ini, dikombinasikan dengan fakta bahwa algoritma kompresi modern seperti JPEG dan H.264 juga menerapkan DCT dalam kerangka kerjanya, menjadikan DCT sebagai transformasi domain frekuensi yang lazim dipilih untuk analisis citra.

2.7 Frequency-Aware Decomposition (FAD)

Frequency-Aware Decomposition (FAD) adalah metode dekomposisi citra dalam domain frekuensi yang diperkenalkan oleh Qian et al. [26] sebagai bagian dari kerangka kerja *Frequency in Face Forgery Network* (F^3-Net). FAD bertujuan untuk mengungkap pola pemalsuan (*forgery patterns*) dengan cara membagi citra masukan ke dalam beberapa komponen frekuensi. Berbeda dengan studi terdahulu yang menggunakan *filter bank* buatan tangan (*hand-crafted*) dengan konfigurasi tetap sehingga gagal mencakup keseluruhan domain frekuensi dan tidak mampu menangkap pola pemalsuan secara adaptif, FAD membagi citra dalam domain frekuensi berdasarkan sekumpulan *filter* frekuensi yang dapat dipelajari (*learnable*) [26].

Untuk membagi spektrum frekuensi, FAD merancang N buah *filter* dasar biner $\{f^i_{base}\}^N_{i=1}$ (disebut juga *mask*) yang secara eksplisit membagi domain

frekuensi ke dalam pita frekuensi rendah, menengah, dan tinggi. Pada setiap *filter* dasar tersebut ditambahkan sebuah *filter* yang dapat dipelajari $\{f_w^i\}_{i=1}^N$, sehingga proses penyaringan frekuensi dilakukan melalui perkalian elemen per elemen antara respons frekuensi citra masukan dengan kombinasi *filter* $f_{base}^i + \sigma(f_w^i)$. Dengan demikian, untuk sebuah citra masukan x, komponen citra hasil dekomposisi diperoleh sebagaimana ditunjukkan pada Persamaan 2.8.

$$y_i = \mathscr{D}^{-1}\left\{\mathscr{D}(x) \odot \left[f_{base}^i + \sigma(f_w^i)\right]\right\}, \quad i = \{1,\ldots,N\} \tag{2.8}$$

dengan $\mathscr{D}$ menyatakan operasi *Discrete Cosine Transform* (DCT), $\mathscr{D}^{-1}$ menyatakan *Inverse Discrete Cosine Transform* (IDCT), dan $\odot$ menyatakan perkalian elemen per elemen (*element-wise product*). Fungsi $\sigma(\cdot)$ berfungsi untuk membatasi (*squeeze*) nilai *filter* yang dapat dipelajari ke dalam rentang antara -1 dan $+1$, sebagaimana didefinisikan pada Persamaan 2.9.

$$\sigma(x) = \frac{1-\exp(-x)}{1+\exp(-x)} \tag{2.9}$$

DCT dipilih sebagai transformasi domain frekuensi pada FAD karena penggunaannya yang luas pada pemrosesan citra serta tata letak distribusi frekuensinya yang teratur, yaitu respons frekuensi rendah ditempatkan pada sudut kiri atas dan respons frekuensi tinggi pada sudut kanan bawah. Selain itu, algoritma kompresi modern seperti JPEG dan H.264 juga menerapkan DCT dalam kerangka kerjanya, sehingga FAD berbasis DCT lebih kompatibel dalam mendeskripsikan *artifact* kompresi yang menyertai pola pemalsuan [26].

Pengamatan terhadap spektrum daya (*power spectrum*) DCT dari citra natural menunjukkan bahwa distribusi spektralnya tidak seragam, dengan sebagian besar amplitudo terkonsentrasi pada area frekuensi rendah. Oleh karena itu, *filter* dasar f_{base} digunakan untuk membagi spektrum menjadi N pita dengan energi yang kurang lebih setara, dari frekuensi rendah ke frekuensi tinggi, sementara penambahan *filter* yang dapat dipelajari $\{f_w^i\}_{i=1}^N$ memberikan kemampuan adaptasi yang lebih baik dalam memilih pita frekuensi yang relevan di luar *filter* dasar yang tetap. Secara empiris, Qian et al. [26] menetapkan jumlah pita $N = 3$, dengan pita frekuensi rendah f_{base}^1 mencakup $1/16$ pertama dari keseluruhan spektrum, pita frekuensi menengah f_{base}^2 berada di antara $1/16$ dan $1/8$ spektrum, dan pita frekuensi tinggi f_{base}^3 mencakup $7/8$ sisanya.

Komponen-komponen citra hasil dekomposisi tersebut kemudian disusun bertumpuk sepanjang sumbu *channel* dan diteruskan ke sebuah *Convolutional Neural Network* (CNN) untuk menggali pola pemalsuan secara komprehensif [26]. Penelitian ini mengadopsi FAD dengan satu penyesuaian, yaitu penambahan satu jalur spektrum penuh (all) di samping ketiga pita F³-Net, sebagaimana dirinci pada Bab 3.

2.8 Dataset

2.8.1 FaceForensics++

FaceForensics++ (FF++) [25] adalah dataset forensik wajah berskala besar yang dirilis pada tahun 2019. Dataset ini memuat 1.000 video asli yang bersumber dari 977 video YouTube, yang kemudian dimanipulasi menggunakan empat metode awal, yaitu DeepFakes, Face2Face, FaceSwap, dan NeuralTextures, serta satu metode tambahan, FaceShifter, yang disertakan pada pembaruan tahun 2020. Seluruh video tersedia dalam tiga tingkat kompresi. Dataset ini merupakan tolok ukur utama (*primary benchmark*) di bidang deteksi *deepfake* dan digunakan sebagai dataset pelatihan dalam penelitian ini.

2.8.2 Celeb-DF v2

Celeb-DF v2 [32] adalah dataset *deepfake* berskala besar yang merupakan versi pembaruan dari Celeb-DF v1 (2019) dan dirilis pada tahun 2020. Dataset ini dirancang sebagai tolok ukur yang menantang dengan menyediakan *deepfake* berkualitas visual tinggi, karena dataset terdahulu memiliki artefak yang relatif mudah terlihat. Dataset ini memuat 590 video asli dari tokoh terkenal (*celebrities*) dan 300 video asli tambahan dari YouTube, serta 5.639 video *deepfake* yang dihasilkan melalui proses sintesis yang disempurnakan, termasuk koreksi warna untuk mengurangi ketidakcocokan pada batas penggabungan wajah.

2.8.3 DeepFake Detection Dataset (DFD)

DeepFake Detection Dataset (DFD) merupakan dataset yang dirilis pada tahun 2019, dikurasi oleh Google dan Jigsaw serta didistribusikan melalui repositori FaceForensics++ [25]. Dataset ini memuat 363 video asli dari sejumlah aktor yang memberikan persetujuan (*consent*) dan 3.068 video *deepfake* berkualitas tinggi

dalam berbagai skenario. Sebagaimana Celeb-DF v2, DFD digunakan dalam penelitian ini hanya untuk evaluasi generalisasi *cross-dataset*, tanpa diikutsertakan dalam proses pelatihan maupun validasi.

2.8.4 DeepFakeFace Dataset (DFF)

DeepFakeFace (DFF) [33] merupakan dataset yang dirilis pada tahun 2023 untuk mendukung penelitian deteksi wajah sintetis, dengan fokus pada citra wajah yang dihasilkan oleh model *diffusion* modern. Dataset ini memuat citra wajah asli dari selebritas (bersumber dari IMDB-WIKI) yang dipasangkan dengan citra wajah palsu yang dibangkitkan menggunakan beberapa generator berbasis *diffusion* dan GAN, seperti Stable Diffusion, InsightFace, serta model sejenis. Berbeda dari dataset berbasis *deepfake* video, DFF menyediakan data dalam bentuk citra statis, sehingga cocok untuk mengevaluasi ketahanan model terhadap manipulasi wajah yang dihasilkan secara *generative*. Dalam penelitian ini, DFF digunakan hanya untuk evaluasi generalisasi *cross-dataset*, tanpa diikutsertakan dalam proses pelatihan maupun validasi.

2.9 Metrik Evaluasi

Untuk menilai kinerja model dalam membedakan wajah asli dari hasil manipulasi, digunakan beberapa metrik evaluasi standar pada tugas klasifikasi biner. Seluruh metrik dihitung berdasarkan empat komponen dasar pada *confusion matrix*, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN).

Accuracy mengukur proporsi prediksi yang benar terhadap keseluruhan sampel, sebagaimana ditunjukkan pada Persamaan 2.10. Metrik ini memberikan gambaran umum performa model, tetapi kurang representatif pada data dengan distribusi kelas yang tidak seimbang.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.10}$$

Sebagai ilustrasi, apabila suatu dataset memuat 83% sampel *fake* dan 17% sampel *real* (rasio sekitar 1:5), sebuah model yang memprediksi seluruh sampel sebagai *fake* akan memperoleh *accuracy* 83%. Nilai ini tampak tinggi, padahal model tersebut tidak mampu mengenali satu pun sampel *real*. *Accuracy* yang tinggi tersebut semata-mata berasal dari dominasi kelas mayoritas, bukan dari kemampuan

model membedakan kedua kelas. Oleh karena itu, diperlukan metrik yang menilai performa per kelas secara lebih adil.

Salah satunya adalah *F1-Score*, yaitu rata-rata harmonik antara *precision* dan *recall*, sebagaimana ditunjukkan pada Persamaan 2.11. Karena menggabungkan kedua aspek tersebut, F1-Score lebih andal dibandingkan *accuracy* ketika jumlah sampel antarkelas timpang, sebab penalti diberikan ketika salah satu dari *precision* atau *recall* bernilai rendah.

$$F1\text{-}Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{2.11}$$

dengan $Precision = \frac{TP}{TP+FP}$ dan $Recall = \frac{TP}{TP+FN}$.

Selain menilai performa pada satu ambang keputusan, kemampuan diskriminatif model juga diukur secara menyeluruh melalui ROC-AUC. ROC-AUC adalah luas area di bawah kurva *Receiver Operating Characteristic* (ROC), yaitu kurva yang memetakan hubungan antara *True Positive Rate* (TPR) terhadap *False Positive Rate* (FPR) pada berbagai nilai ambang klasifikasi. Kedua nilai tersebut didefinisikan pada Persamaan 2.12, sedangkan ROC-AUC didefinisikan pada Persamaan 2.13.

$$TPR = \frac{TP}{TP+FN}, \quad FPR = \frac{FP}{FP+TN} \tag{2.12}$$

$$ROC\text{-}AUC = \int_0^1 TPR\, d(FPR) \tag{2.13}$$

Nilai ROC-AUC berkisar antara 0 hingga 1; semakin mendekati 1, semakin baik kemampuan model dalam memisahkan kelas asli dan manipulasi tanpa bergantung pada pemilihan ambang tertentu.

ROC-AUC dan F1-Score sama-sama lebih sesuai untuk data timpang dibandingkan *accuracy*, namun mengukur aspek yang berbeda sehingga digunakan secara bersama. ROC-AUC menilai kemampuan diskriminatif model secara menyeluruh dan independen terhadap ambang, sedangkan F1-Score menilai keseimbangan *precision-recall* pada satu ambang keputusan. Karena model dapat memperoleh ROC-AUC yang tinggi namun F1-Score yang rendah ketika keseimbangan pada ambang belum optimal, keduanya dijadikan metrik utama, sedangkan *accuracy* hanya berperan sebagai metrik pendukung.