BAB 2
LANDASAN TEORI

2.1 *Facial Expression Recognition*

Facial Expression Recognition (FER) merupakan bidang dalam *computer vision* yang mempelajari pengenalan ekspresi wajah secara otomatis dari citra atau video. Dalam klasifikasi berbasis citra, sistem menerima citra wajah sebagai masukan dan menghasilkan label ekspresi sebagai keluaran. FER tidak menentukan keseluruhan keadaan psikologis seseorang, melainkan mengenali pola visual ekspresi yang tampak pada wajah agar dapat diproses oleh komputer [2, 9].

Perubahan ekspresi dapat terlihat pada beberapa bagian wajah, antara lain alis, mata, pipi, dan mulut. Tujuh kategori yang digunakan pada subset ekspresi dasar RAF-DB adalah marah (*angry*), jijik (*disgust*), takut (*fear*), senang (*happy*), sedih (*sad*), terkejut (*surprise*), dan netral (*neutral*) [5, 8]. Dalam praktiknya, ekspresi yang berbeda dapat memiliki ciri lokal yang serupa. Sebagai contoh, perubahan di sekitar mata atau mulut dapat muncul pada lebih dari satu kelas. Oleh karena itu, model perlu mempelajari kombinasi ciri dari beberapa area wajah, bukan hanya satu bagian.

FER pada citra *in-the-wild* lebih sulit daripada FER pada lingkungan terkontrol. Pose kepala, pencahayaan, oklusi, latar belakang, usia, identitas, kualitas citra, dan ambiguitas ekspresi dapat berubah antarsampel [6, 7]. CNN banyak digunakan pada FER karena dapat mempelajari ciri visual secara bertingkat, mulai dari tepi dan tekstur sampai representasi bagian wajah yang lebih kompleks [3, 4].

2.2 Dataset RAF-DB

Real-world Affective Faces Database (RAF-DB) merupakan dataset ekspresi wajah yang dihimpun dari citra dunia nyata. RAF-DB memiliki subset ekspresi dasar dan subset ekspresi majemuk. Penelitian ini menggunakan subset ekspresi dasar karena keluaran model dibatasi pada tujuh kelas ekspresi. Subset tersebut memiliki pembagian resmi sebanyak 12.271 citra latih dan 3.068 citra uji [5, 8].

Karakteristik in-the-wild membuat RAF-DB memuat variasi pose, pencahayaan, oklusi, dan karakteristik subjek yang tidak seragam. Variasi ini sesuai untuk menguji kemampuan generalisasi Facial Expression Recognition

(FER), tetapi sekaligus meningkatkan kesulitan klasifikasi. Selain variasi visual, distribusi kelas RAF-DB juga tidak seimbang secara ekstrem. Untuk memetakan ketimpangan tersebut sebelum melakukan eksperimen, distribusi data latih resmi yang digunakan dalam penelitian ini ditunjukkan pada Tabel 2.1.

Tabel 2.1. Distribusi kelas pada data latih resmi RAF-DB [1].

Kelas	**Jumlah sampel**	**Posisi distribusi**
Angry	705	Minoritas
Disgust	717	Minoritas
Fear	281	Minoritas terkecil
Happy	4.772	Mayoritas terbesar
Sad	1.982	Menengah
Surprise	1.290	Menengah
Neutral	2.524	Menengah
Total	**12.271**	

Berdasarkan data pada Tabel 2.1, ketimpangan jumlah sampel antar-kelas terlihat sangat mencolok pada total 12.271 citra data latih. Kelas *happy* mendominasi sebagai mayoritas terbesar dengan 4.772 citra, sementara kelas *fear* bertindak sebagai minoritas terkecil dengan hanya memiliki 281 citra. Kondisi ini menunjukkan adanya kesenjangan ekstrem di mana model berpeluang menerima paparan contoh ekspresi *happy* hampir 17 kali lebih sering daripada ekspresi *fear*. Di sisi lain, kelas-kelas seperti *sad*, *surprise*, dan *neutral* menempati posisi distribusi menengah, sedangkan kelas *angry* (705 sampel) dan *disgust* (717 sampel) memperlebar deretan kelas minoritas yang rentan terabaikan saat proses optimasi.

2.3 Representasi Citra RGB dan Penskalaan Piksel

Citra digital berwarna dapat direpresentasikan sebagai tensor $H \times W \times C$, dengan H sebagai tinggi, W sebagai lebar, dan C sebagai jumlah kanal. Pada citra RGB, $C = 3$ dan ketiga kanal tersebut merepresentasikan merah, hijau, dan biru. Dengan demikian, citra berukuran 100×100 piksel direpresentasikan sebagai tensor $100 \times 100 \times 3$.

Nilai kanal pada citra delapan bit berada pada rentang 0–255. Nilai tersebut dapat dikonversi ke tipe *floating point* pada rentang 0–1 melalui

$$I_{\text{skala}}(x,y,c) = \frac{I(x,y,c)}{255}. \tag{2.1}$$

Penskalaan ini hanya mengubah rentang numerik yang diterima jaringan. Ketiga kanal RGB tetap dipertahankan sehingga proses tersebut tidak mengubah citra menjadi *grayscale*. Penskalaan piksel juga berbeda dari *batch normalization*: penskalaan dilakukan pada citra masukan, sedangkan *batch normalization* bekerja pada aktivasi di dalam jaringan.

2.4 *Convolutional Neural Network*

2.4.1 Konsep Dasar CNN

Convolutional Neural Network (CNN) merupakan jaringan saraf yang dirancang untuk memproses data berbentuk grid seperti citra. CNN menggunakan filter dengan bobot bersama untuk mempelajari pola lokal pada berbagai posisi. Susunan beberapa lapisan memungkinkan jaringan membentuk representasi hierarkis: lapisan awal menangkap pola sederhana, sedangkan lapisan lebih dalam menggabungkannya menjadi ciri yang lebih abstrak [3, 4].

2.4.2 *Convolutional Layer* dan ReLU

Lapisan konvolusi menggeser kernel pada citra atau *feature map* untuk menghasilkan *feature map* baru. Secara sederhana, operasi pada satu posisi dapat dituliskan sebagai

$$S(i,j) = \sum_m \sum_n I(i+m, j+n)K(m,n) + b, \tag{2.2}$$

dengan I sebagai masukan, K sebagai kernel, b sebagai bias, dan S sebagai keluaran. Ukuran kernel, jumlah filter, *stride*, dan *padding* menentukan bentuk serta kapasitas lapisan.

Setelah konvolusi, fungsi aktivasi menambahkan non-linearitas. ReLU didefinisikan sebagai

$$f(x) = \max(0,x). \tag{2.3}$$

ReLU meneruskan nilai positif dan mengubah nilai negatif menjadi nol. Tanpa aktivasi non-linear, rangkaian lapisan hanya membentuk transformasi linear dan tidak cukup untuk mempelajari hubungan visual yang kompleks.

2.4.3 *Pooling* dan *Global Average Pooling*

Max pooling mereduksi dimensi spasial dengan memilih nilai maksimum dalam jendela lokal. Reduksi ini menurunkan beban komputasi dan membuat representasi lebih toleran terhadap pergeseran kecil. Pada bagian akhir jaringan, *Global Average Pooling* (GAP) menghitung rata-rata setiap *feature map*:

$$g_c = \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} X_{i,j,c}, \tag{2.4}$$

dengan g_c sebagai keluaran kanal ke-c. GAP menghasilkan satu nilai untuk setiap kanal dan membutuhkan parameter klasifikasi lebih sedikit daripada meratakan seluruh *feature map* menggunakan *flatten* [16].

2.4.4 *Batch Normalization* dan *Dropout*

Batch normalization menormalkan aktivasi pada satu *mini-batch*, kemudian menerapkan skala dan pergeseran yang dapat dipelajari:

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}, \qquad y_i = \gamma \hat{x}_i + \beta. \tag{2.5}$$

Parameter μ_B dan σ_B^2 merupakan rata-rata dan varians *batch*, sedangkan γ dan β dipelajari selama pelatihan. Teknik ini membantu menstabilkan optimasi, tetapi bukan metode penyeimbangan kelas [17].

Dropout menonaktifkan sebagian unit secara acak selama pelatihan. Dengan probabilitas *dropout* p, sebuah unit tidak digunakan pada iterasi tertentu. Mekanisme ini mengurangi ketergantungan jaringan pada kombinasi unit tertentu dan digunakan sebagai regularisasi untuk menekan *overfitting* [18].

2.4.5 *Dense*, *Softmax*, dan Fungsi Kerugian

Lapisan *dense* menggabungkan vektor fitur menjadi skor kelas atau *logit*. Untuk K kelas, *softmax* mengubah logit z_j menjadi probabilitas:

$$\hat{y}_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}. \tag{2.6}$$

Jumlah seluruh probabilitas adalah satu dan kelas dengan probabilitas terbesar dipilih sebagai prediksi.

Untuk label *one-hot*, perbedaan prediksi dan label sebenarnya dihitung menggunakan *categorical cross-entropy*:

$$L = -\sum_{k=1}^{K} y_k \log(\hat{y}_k). \tag{2.7}$$

Jika probabilitas kelas yang benar kecil, nilai kerugian menjadi besar. Pada data tidak seimbang, kelas mayoritas muncul lebih sering dan dapat memberikan kontribusi kumulatif lebih besar terhadap kerugian. Karena itu, distribusi sampel atau bobot kerugian perlu diperhatikan.

Bobot jaringan diperbarui oleh algoritma optimasi. Adam menggunakan estimasi momen pertama dan kedua gradien untuk menyesuaikan besar pembaruan parameter [19]. *Learning rate* mengatur besar langkah pembaruan; nilai terlalu besar dapat membuat pelatihan tidak stabil, sedangkan nilai terlalu kecil dapat memperlambat konvergensi. Oleh sebab itu, *learning rate* merupakan salah satu *hyperparameter* yang perlu dipilih menggunakan data validasi.

2.5 *Class Imbalance*

2.5.1 Pengertian dan Dampak *Class Imbalance*

Class imbalance adalah kondisi ketika jumlah sampel antarkelas tidak setara. Besarnya ketimpangan dapat diringkas menggunakan *imbalance ratio* (IR):

$$\mathrm{IR} = \frac{\max_i n_i}{\min_i n_i}, \tag{2.8}$$

dengan n_i sebagai jumlah sampel kelas ke-i. Nilai IR hanya menunjukkan perbandingan kelas terbesar dan terkecil; distribusi seluruh kelas tetap perlu dilaporkan agar bentuk ketimpangan tidak disederhanakan menjadi satu angka.

Pada pelatihan standar, sampel kelas mayoritas lebih sering berkontribusi pada pembaruan bobot. Akibatnya, model dapat membentuk batas keputusan yang lebih menguntungkan kelas mayoritas dan memiliki *recall* rendah pada kelas minoritas. Akurasi keseluruhan masih dapat terlihat tinggi apabila sebagian besar data uji berasal dari kelas yang mudah atau dominan [20, 21]. Pada RAF-DB, dampak tersebut perlu diperiksa khususnya pada *fear*, *disgust*, dan *angry* [9, 22].

2.5.2 Pendekatan Penanganan Ketidakseimbangan

Penanganan ketidakseimbangan dapat dilakukan pada tingkat data maupun tingkat algoritma [20]. Penelitian ini berfokus pada pendekatan tingkat data agar kontribusi pemerataan paparan dan variasi augmentasi dapat diamati secara langsung. Untuk memahami karakteristik, kelebihan, serta kelemahan dari masing-masing metode penanganan data tidak seimbang tersebut, perbandingan pendekatan tingkat data yang relevan ditunjukkan pada Tabel 2.2.

Tabel 2.2. Perbandingan pendekatan penanganan ketidakseimbangan kelas.

Pendekatan	**Prinsip**	**Kelebihan**	**Risiko/keterbatasan**
Undersampling	Mengurangi sampel kelas mayoritas	Menurunkan dominasi dan biaya komputasi	Membuang informasi dari kelas mayoritas
Random oversampling	Mengulang sampel kelas minoritas	Mempertahankan seluruh data asli dan menyeimbangkan paparan	Pengulangan identik dapat meningkatkan risiko *overfitting*
Balanced sampling/batch	Mengatur jumlah sampel setiap kelas dalam pengambilan data atau *batch*	Mengendalikan kontribusi kelas pada pembaruan bobot	Tidak menciptakan informasi visual baru
Class weighting	Memberi bobot kerugian lebih besar pada kelas yang jarang	Tidak menggandakan data	Bobot besar dapat membuat optimasi lebih sensitif terhadap sampel sulit atau label salah
Augmentasi kelas minoritas	Membuat variasi transformasi dari sampel yang tersedia	Mengurangi pengulangan tampilan identik	Tidak menjamin jumlah paparan seimbang jika digunakan sendiri

Buda dkk. membandingkan beberapa cara penanganan ketidakseimbangan pada CNN dan menunjukkan bahwa *oversampling* efektif dalam banyak kondisi yang mereka uji [21]. Meskipun demikian, penyeimbangan jumlah baris dataset belum otomatis menjamin komposisi setiap *batch*. Jika pengambilan dilakukan secara acak tanpa kendali, beberapa *batch* masih dapat didominasi kelas tertentu. *Balanced batch* mengendalikan paparan dengan memasukkan jumlah sampel yang

sama dari setiap kelas ke dalam satu *batch*.

2.5.3 Hubungan *Oversampling*, *Balanced Batch*, dan Augmentasi

Oversampling dan augmentasi memiliki fungsi berbeda. *Oversampling* mengatur frekuensi kemunculan kelas, sedangkan augmentasi menambah keragaman tampilan citra. Augmentasi pada distribusi yang tetap didominasi kelas mayoritas tidak membuktikan bahwa setiap kelas memperoleh paparan seimbang. Sebaliknya, *oversampling* tanpa variasi dapat mengulang citra minoritas yang sama berkali-kali.

Kombinasi keduanya dapat digunakan sebagai pendekatan tingkat data: *oversampling* membentuk jumlah kelas yang setara, *balanced batch* menjaga kesetaraan paparan pada setiap pembaruan, dan augmentasi mengubah salinan tambahan agar tidak seluruhnya identik. Dengan demikian, augmentasi tidak berdiri sebagai satu-satunya solusi, tetapi bekerja setelah jumlah dan paparan kelas dikendalikan.

2.6 *Data Augmentation*

2.6.1 Pengertian dan Prinsip *Label-Preserving*

Data augmentation menambah variasi data latih dengan menerapkan transformasi pada citra yang tersedia. Transformasi yang digunakan harus *label-preserving*, yaitu tidak mengubah identitas kelas ekspresi. Augmentasi dapat membantu generalisasi, tetapi transformasi yang terlalu kuat dapat memotong atau menutupi ciri utama ekspresi dan menimbulkan sampel yang tidak lagi konsisten dengan label [10, 23]. Karena itu, jenis, peluang, dan intensitas transformasi perlu dipilih melalui data validasi serta diperiksa secara visual.

Augmentasi hanya diterapkan pada data latih. Data validasi dan uji tidak diaugmentasi agar tetap merepresentasikan citra asli dan dapat digunakan untuk membandingkan konfigurasi secara adil.

2.6.2 Kategori dan Teknik Augmentasi

Shorten dan Khoshgoftaar membagi augmentasi citra ke dalam transformasi geometrik, ruang warna atau fotometrik, erasing, pencampuran citra, ruang fitur, dan pendekatan generatif [10]. Penelitian ini memusatkan pengujian pada

transformasi geometrik, fotometrik, dan erasing/oklusi karena ketiganya berkaitan langsung dengan variasi alami ekspresi manusia in-the-wild pada dataset RAF-DB. Untuk memetakan hubungan antara kategori, teknik yang dipilih, serta batasan penerapannya dalam eksperimen ini, rincian lengkapnya disajikan pada Tabel 2.3.

Tabel 2.3. Kategori augmentasi yang digunakan dalam penelitian.

Kategori	**Teknik**	**Variasi yang disimulasikan**	**Risiko jika berlebihan**
Geometrik	*Horizontal flip*, rotasi, pergeseran, *zoom*, dan *RandomCrop*	Arah, kemiringan, posisi, skala, dan komposisi wajah	Bagian penting wajah terpotong atau geometri menjadi tidak wajar
Fotometrik	Penyesuaian kecerahan dan *ColorJitter*	Pencahayaan, kontras, saturasi, dan rona	Warna atau kontras menjadi tidak realistis
Erasing/oklusi	*CoarseDropout*	Bagian wajah tertutup secara parsial	Ciri ekspresi utama hilang

Merujuk pada Tabel 2.3, pengelompokan augmentasi ini dirancang untuk memaksa model mengenali fitur ekspresi yang robust terhadap distorsi dunia nyata. Kategori geometrik seperti horizontal flip, rotasi, hingga RandomCrop difokuskan untuk merekayasa variasi spasial wajah. Namun, sebagaimana dirangkum dalam kolom risiko Tabel 2.3, penerapan parameter geometrik yang terlalu agresif berpotensi merusak struktur spasial atau memotong komponen visual esensial seperti mata dan mulut. Sementara itu, kategori fotometrik mengondisikan model agar tidak bergantung pada kondisi pencahayaan tertentu lewat simulasi fluktuasi kecerahan dan rona warna. Risiko utama dari manipulasi ini adalah munculnya citra artifisial yang tidak realistis jika intensitas jitter terlalu tinggi.

Horizontal flip membalik citra pada sumbu vertikal dan sesuai untuk menambah variasi arah wajah. Rotasi, pergeseran, dan *zoom* mensimulasikan variasi posisi serta skala. *RandomCrop* memilih area citra secara acak lalu mengembalikannya ke ukuran masukan; teknik ini dapat mengurangi ketergantungan model pada posisi wajah yang identik, tetapi skala potongan perlu dibatasi agar bagian ekspresif tetap terlihat [12].

Penyesuaian kecerahan mengubah intensitas ketiga kanal secara terkendali. *ColorJitter* dapat mengubah kecerahan, kontras, saturasi, dan rona secara acak

sehingga mewakili variasi fotometrik yang lebih luas. Transformasi ini tetap bekerja pada citra RGB dan tidak mengubah jumlah kanal.

CoarseDropout menutup satu atau beberapa area persegi panjang pada citra. Secara konsep, teknik ini berkaitan dengan *Random Erasing*, yang digunakan untuk meningkatkan ketahanan model terhadap oklusi [11]. Ukuran dan jumlah area harus dibatasi agar transformasi tidak menghilangkan sebagian besar ciri ekspresi. Kombinasi *RandomCrop*, *ColorJitter*, dan teknik *erasing* juga telah digunakan pada penelitian FER, tetapi sering digabungkan dengan arsitektur atau mekanisme lain sehingga kontribusi masing-masing transformasi tidak selalu dapat dipisahkan [12, 13].

2.7 Pencarian *Hyperparameter*

Parameter adalah bobot yang dipelajari model selama pelatihan, sedangkan *hyperparameter* ditentukan sebelum atau di luar proses pembaruan bobot. Contohnya meliputi *learning rate*, *batch size*, jumlah filter, jumlah unit *dense*, dan *dropout rate*. Nilai tersebut memengaruhi kapasitas, regularisasi, kestabilan optimasi, serta kebutuhan komputasi sehingga tidak seharusnya dipilih hanya berdasarkan dugaan.

Grid search mengevaluasi seluruh kombinasi pada kisi nilai yang ditentukan. Jumlah percobaannya tumbuh sebagai hasil perkalian jumlah kandidat setiap parameter. *Random search* mengambil kombinasi secara acak dari ruang yang sama. Bergstra dan Bengio menunjukkan bahwa *random search* dapat lebih efisien ketika hanya sebagian dimensi parameter berpengaruh besar, karena percobaan acak menjelajahi lebih banyak nilai berbeda pada setiap dimensi [24].

Anggaran jumlah percobaan tetap harus ditentukan berdasarkan sumber daya komputasi. Setiap konfigurasi dilatih menggunakan data latih dan dibandingkan pada data validasi menggunakan metrik yang ditetapkan sebelumnya. Data uji tidak digunakan untuk memilih parameter. Prinsip yang sama berlaku pada pemilihan intensitas augmentasi: rentang kandidat harus cukup aman untuk mempertahankan label, sedangkan konfigurasi terpilih ditentukan oleh hasil validasi, bukan hasil uji.

2.8 Pembagian Data, Kebocoran Data, dan Reproduksibilitas

Data latih digunakan untuk memperbarui bobot model, data validasi digunakan untuk memilih konfigurasi dan menghentikan pelatihan, sedangkan data uji digunakan satu kali pada evaluasi akhir. Pembagian terstratifikasi menjaga proporsi kelas pada data latih dan validasi. Namun, stratifikasi tidak menyeimbangkan kelas; teknik tersebut hanya mempertahankan proporsi distribusi asal.

Data leakage terjadi ketika informasi yang seharusnya hanya tersedia pada tahap evaluasi memengaruhi pelatihan atau pemilihan model. Contohnya adalah memilih parameter berdasarkan hasil data uji, menerapkan *oversampling* sebelum pembagian sehingga salinan satu citra berada di dua bagian data, atau membiarkan citra identik tersebar pada data latih dan uji. Kebocoran dapat menghasilkan estimasi performa yang terlalu optimistis. Oleh karena itu, audit duplikat, pembagian data, serta penyeimbangan harus dilakukan dalam urutan yang terkontrol.

Pelatihan CNN bersifat stokastik karena inisialisasi bobot, pengacakan sampel, dan augmentasi dapat berubah. *Random seed* membuat rangkaian acak dapat direproduksi, tetapi satu *seed* belum menunjukkan kestabilan metode. Pengulangan pada beberapa *seed* memungkinkan pelaporan rata-rata dan simpangan baku sehingga perbedaan skenario tidak hanya didasarkan pada satu hasil pelatihan.

2.9 Metrik Evaluasi

2.9.1 *Confusion Matrix* dan Metrik per Kelas

Confusion matrix menyandingkan kelas sebenarnya dengan kelas prediksi. Untuk kelas ke-i, evaluasi multikelas dapat dibaca dengan pendekatan satu-lawan-semua: TP_i adalah sampel kelas i yang diprediksi benar, FP_i adalah sampel kelas lain yang diprediksi sebagai i, dan FN_i adalah sampel kelas i yang diprediksi sebagai kelas lain. Berdasarkan nilai tersebut, *precision*, *recall*, dan F1-score dirumuskan

sebagai [25]

$$\begin{aligned} Precision_i &= \frac{TP_i}{TP_i + FP_i}, \\ Recall_i &= \frac{TP_i}{TP_i + FN_i}, \\ F1_i &= 2\frac{Precision_i Recall_i}{Precision_i + Recall_i}. \end{aligned} \tag{2.9}$$

Precision menunjukkan ketepatan prediksi suatu kelas, sedangkan *recall* menunjukkan proporsi sampel kelas tersebut yang berhasil dikenali. Pada masalah ketidakseimbangan, *recall* kelas minoritas penting karena memperlihatkan berapa banyak sampel minoritas yang terlewat. F1-score menyeimbangkan *precision* dan *recall*.

Confusion matrix mentah menunjukkan jumlah kesalahan aktual. Normalisasi per baris membagi setiap elemen dengan jumlah sampel pada kelas sebenarnya sehingga setiap baris berjumlah satu. Bentuk ternormalisasi memudahkan perbandingan pola kesalahan antarkelas yang jumlah sampelnya berbeda, tetapi tidak menggantikan matriks mentah.

2.9.2 Metrik Global pada Data Tidak Seimbang

Akurasi menghitung proporsi seluruh prediksi yang benar:

$$Accuracy = \frac{\sum_{i=1}^{K} TP_i}{N}. \tag{2.10}$$

Pada data tidak seimbang, akurasi dapat didominasi kelas dengan jumlah sampel besar. Karena itu, metrik agregat yang memberi perhatian pada setiap kelas perlu dilaporkan bersama akurasi [20, 25].

Macro precision, *macro recall*, dan *macro F1-score* menghitung rata-rata tanpa bobot:

$$\begin{aligned} MacroPrecision &= \frac{1}{K}\sum_{i=1}^{K} Precision_i, \\ MacroRecall &= \frac{1}{K}\sum_{i=1}^{K} Recall_i, \\ MacroF1 &= \frac{1}{K}\sum_{i=1}^{K} F1_i. \end{aligned} \tag{2.11}$$

Setiap kelas memiliki kontribusi yang sama, sehingga *macro F1-score* sesuai

sebagai metrik utama ketika performa kelas minoritas tidak boleh tertutup oleh kelas mayoritas.

Untuk klasifikasi multikelas satu-label, *balanced accuracy* merupakan rata-rata *recall* seluruh kelas dan nilainya sama dengan *macro recall*. Sementara itu, *weighted F1-score* menggunakan jumlah sampel kelas sebagai bobot:

$$WeightedF1 = \sum_{i=1}^{K} \frac{n_i}{N} F1_i. \tag{2.12}$$

Metrik ini tetap mencerminkan distribusi alami data uji sehingga perlu dibaca bersama *macro F1-score*.

Pemerataan *recall* dapat diringkas menggunakan *recall gap*:

$$G_{\text{recall}} = \max_i(Recall_i) - \min_i(Recall_i). \tag{2.13}$$

Nilai lebih kecil menunjukkan jarak kemampuan pengenalan antarkelas yang lebih sempit. Namun, nilai tersebut tidak boleh dibaca sendiri karena seluruh *recall* dapat sama-sama rendah. Oleh karena itu, *recall gap* harus dianalisis bersama *macro F1-score* dan metrik per kelas.

2.9.3 Rata-rata dan Simpangan Baku Antarseed

Jika satu skenario dijalankan pada R *seed*, rata-rata metrik x_r dihitung sebagai

$$\bar{x} = \frac{1}{R} \sum_{r=1}^{R} x_r, \tag{2.14}$$

sedangkan simpangan baku sampel dihitung sebagai

$$s = \sqrt{\frac{1}{R-1} \sum_{r=1}^{R} (x_r - \bar{x})^2}. \tag{2.15}$$

Rata-rata menggambarkan kecenderungan performa, sedangkan simpangan baku menunjukkan variasi antarpelatihan. Skenario yang memiliki rata-rata tinggi tetapi simpangan baku besar perlu ditafsirkan lebih hati-hati daripada skenario dengan performa yang konsisten.

2.10 Penelitian Terdahulu

Tabel 2.4 merangkum penelitian yang menjadi dasar pemilihan dataset, penanganan ketidakseimbangan, dan augmentasi.

Tabel 2.4. Ringkasan penelitian terdahulu yang relevan.

Penelitian	**Fokus/metode**	**Hasil atau kontribusi**	**Relevansi dan keterbatasan**
Li dkk. (2017) [5]	RAF-DB dan DLP-CNN	Memperkenalkan dataset ekspresi dasar dan majemuk *in-the-wild*	Menjadi dasar pemilihan dataset, tetapi tidak mengisolasi pengaruh penyeimbangan dan komposisi augmentasi
Buda dkk. (2018) [21]	Studi sistematis ketidakseimbangan pada CNN	Membandingkan *oversampling*, *undersampling*, dan strategi lain	Mendukung penggunaan *oversampling*, tetapi tidak dilakukan pada FER atau RAF-DB
Shorten dan Khoshgoftaar (2019) [10]	Survei augmentasi citra	Mengelompokkan transformasi geometrik, warna, *erasing*, pencampuran, dan generatif	Menjadi dasar kategori augmentasi, tetapi tidak menguji konfigurasi penelitian ini
Zhong dkk. (2020) [11]	*Random Erasing*	Menutupi area acak sebagai regularisasi dan simulasi oklusi	Menjadi dasar konseptual *CoarseDropout*, tetapi bukan studi khusus FER

Penelitian	**Fokus/metode**	**Hasil atau kontribusi**	**Relevansi dan keterbatasan**
Zeng dkk. (2022) [22]	Bias data dan augmentasi spesifik ekspresi	Menganalisis ketimpangan performa kelas FER	Mendukung evaluasi per kelas, tetapi menggunakan pendekatan *meta-learning* yang berbeda
Fan dkk. (2022) [26]	*Class weighting* dan estimasi ketidakpastian	Menangani kelas minoritas pada FER	Mendukung pembanding berbasis bobot, tetapi tidak mengisolasi augmentasi sederhana
Psaroudakis dan Kollias (2022) [14]	MixAugment dan Mixup	Menunjukkan manfaat kombinasi augmentasi untuk FER	Menggunakan augmentasi pencampuran sehingga berbeda dari transformasi sederhana yang diuji
Zhou dkk. (2023) [9]	Fitur global–lokal pada distribusi *long-tailed*	Melaporkan akurasi RAF-DB 89,97%	Menguatkan masalah kelas minoritas, tetapi peningkatan juga dipengaruhi arsitektur khusus
Wu dan Cui (2023) [12]	LA-Net dan *landmark-aware learning*	Menggunakan *crop*, *flip*, *erasing*, dan *color jitter*	Teknik augmentasi relevan, tetapi digabung dengan mekanisme *landmark*

Penelitian	Fokus/metode	Hasil atau kontribusi	Relevansi dan keterbatasan
Yu dkk. (2024) [15]	MixCut	Menggabungkan Mixup, Cutout, dan CutMix pada FER	Menunjukkan komposisi augmentasi penting, tetapi tidak memisahkan tiga kategori yang diuji penelitian ini
Li dkk. (2024) [1]	Augmentasi semantik untuk FER	Menargetkan keragaman kelas minoritas	Mendukung augmentasi minoritas, tetapi menggunakan transformasi ruang fitur yang lebih kompleks
Nemati dkk. (2025) [13]	FER-HA dengan *attention* dan *landmark*	Menggunakan rotasi, *ColorJitter*, dan *CoarseDropout*	Menunjukkan teknik tambahan relevan untuk FER, tetapi tidak diuji pada CNN sederhana yang konstan

2.11 Posisi dan Kesenjangan Penelitian

Penelitian terdahulu menunjukkan tiga hal. Pertama, ketidakseimbangan kelas dapat menurunkan kemampuan CNN mengenali kelas minoritas, sehingga akurasi global tidak cukup. Kedua, *oversampling* mengendalikan frekuensi paparan, tetapi tidak dengan sendirinya menambah variasi visual. Ketiga, augmentasi geometrik, fotometrik, dan oklusi telah digunakan pada FER, tetapi sering menjadi bagian dari arsitektur atau metode kompleks sehingga kontribusi setiap komponen sulit dipisahkan.

Berdasarkan kondisi tersebut, kesenjangan penelitian terletak pada belum jelasnya kontribusi penyeimbangan paparan dan komposisi augmentasi sederhana ketika arsitektur CNN, pembagian data, serta prosedur evaluasi

dipertahankan konstan. Penelitian ini menempatkan distribusi asli sebagai *baseline*, membandingkannya dengan *oversampling* seimbang tanpa augmentasi, kemudian menguji augmentasi umum, *RandomCrop*, *ColorJitter*, dan *CoarseDropout* secara terpisah maupun gabungan.

Kontribusi tidak dinilai hanya dari kenaikan akurasi. Evaluasi menggunakan *macro F1-score*, metrik per kelas, *balanced accuracy*, *recall gap*, distribusi prediksi, *confusion matrix*, dan beberapa *random seed*. Dengan rancangan tersebut, penelitian dapat membedakan apakah perubahan performa berasal dari pemerataan paparan kelas, penambahan variasi visual, atau gabungan keduanya.