

BAB 2

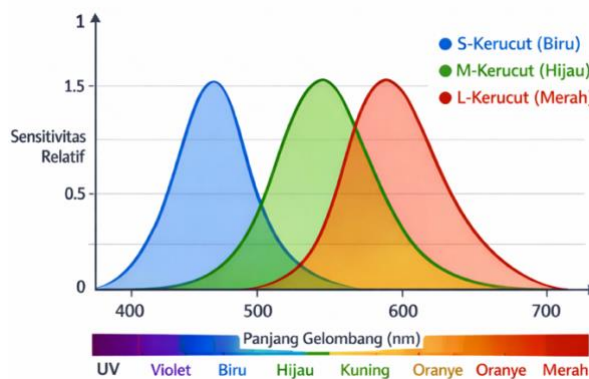
TINJAUAN PUSTAKA

2.1 Landasan Teori

Bagian ini menjelaskan teori yang menjadi dasar dalam perancangan dan penyelesaian permasalahan penelitian. Teori yang dibahas meliputi konsep warna, ruang warna, cat, sistem tinting, *deep learning*, *Convolutional Neural Network*, *transfer learning*, *multi-label classification*, *computer vision*, dan penelitian terdahulu.

2.1.1 Konsep Dasar Warna

Warna merupakan hasil interaksi antara cahaya, objek, dan sistem penglihatan manusia (Wyszecki & Stiles, 2000). Ketika cahaya mengenai suatu benda, sebagian panjang gelombang diserap, sedangkan sebagian lainnya dipantulkan. Cahaya yang dipantulkan diterima oleh retina dan diproses oleh sistem visual sehingga menghasilkan persepsi warna. Warna tidak hanya merupakan fenomena fisik, tetapi juga fenomena perseptual karena persepsi warna dapat berubah bergantung pada kondisi pencahayaan, warna latar belakang, dan adaptasi visual manusia.



Gambar 2.1 Sensitivitas relatif tiga jenis kerucut

Penglihatan warna dijelaskan melalui teori trikromatik dan teori proses oposisi. Teori trikromatik menjelaskan bahwa retina memiliki sel kerucut S (*short*), M (*medium*), dan L (*long*) yang memiliki sensitivitas terhadap rentang panjang gelombang berbeda dan saling tumpang tindih. Persepsi warna dihasilkan dari perbandingan respons ketiga jenis sel tersebut. Sementara itu, teori proses oposisi

menjelaskan pemrosesan warna melalui pasangan merah-hijau, biru-kuning, dan hitam-putih. Kedua teori ini menjadi dasar untuk memahami persepsi warna dan pengembangan ruang warna perseptual, seperti CIELAB (Fairchild, 2013).

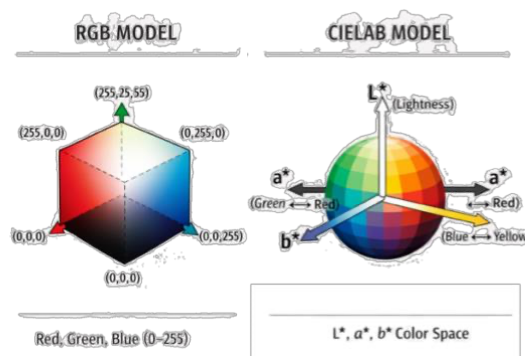
2.1.2 Ruang Warna

Ruang warna (*color space*) merupakan representasi matematis yang mengatur warna dalam sejumlah koordinat numerik. Representasi tersebut memungkinkan warna disimpan, diolah, dan ditampilkan pada berbagai perangkat digital. Setiap ruang warna memiliki karakteristik dan tujuan penggunaan yang berbeda, di antaranya RGB untuk citra digital dan CIELAB untuk pengukuran warna berdasarkan persepsi (Fairchild, 2013).

RGB merupakan model warna aditif yang membentuk warna melalui kombinasi merah (*red*), hijau (*green*), dan biru (*blue*). Model ini digunakan pada kamera digital, monitor, proyektor, serta berbagai sistem pengolahan citra. Pada citra RGB 8-bit, setiap kanal memiliki nilai antara 0 dan 255. Kombinasi (0, 0, 0) merepresentasikan warna hitam, sedangkan (255, 255, 255) merepresentasikan warna putih. Dalam penelitian ini, citra katalog dan foto cat fisik direpresentasikan dalam format RGB. Nilai piksel yang awalnya berada pada rentang 0 – 255 kemudian dinormalisasi ke rentang 0 – 1 sebelum diproses oleh model.

CIELAB merupakan ruang warna yang dikembangkan oleh *Commission Internationale de l'Éclairage* untuk merepresentasikan warna berdasarkan persepsi manusia. CIELAB memiliki tiga komponen, yaitu L^* yang menunjukkan tingkat kecerahan, a^* yang menunjukkan sumbu hijau – merah, dan b^* yang menunjukkan sumbu biru – kuning. Perbedaan antara dua warna dapat dinyatakan menggunakan Delta E (ΔE), di mana nilai yang lebih besar menunjukkan perbedaan warna yang semakin tinggi. Pengukuran CIELAB memerlukan kondisi yang ditetapkan, seperti *illuminant*, *observer*, titik putih acuan, dan kalibrasi alat (Fairchild, 2013).

Pada penelitian ini, RGB digunakan sebagai representasi masukan citra untuk model CNN, sedangkan CIELAB dibahas sebagai konteks pengukuran warna yang umum digunakan dalam industri cat. Model tidak menggunakan nilai CIELAB atau Delta E sebagai fitur masukan maupun metrik evaluasi.



Gambar 2.2 RGB dan CIELAB

2.1.3 Cat

Cat merupakan material pelapis berbentuk cair atau semi-cair yang membentuk lapisan padat setelah diaplikasikan dan dikeringkan. Cat berfungsi memberikan nilai estetika sekaligus melindungi permukaan dari korosi, kelembapan, radiasi ultraviolet, dan paparan bahan kimia. Performa lapisan cat dipengaruhi oleh komposisi bahan, kondisi lingkungan, dan teknik aplikasinya (Lambourne & Strivens, 1999; Müller & Poth, 2017). Secara umum, cat tersusun atas empat komponen utama, yaitu:

- 1) Pigmen, berfungsi memberikan warna, daya tutup, dan sifat tertentu pada lapisan cat.
- 2) Pengikat (*binder* atau resin), yang membentuk lapisan, mengikat partikel pigmen, dan membantu cat melekat pada permukaan.
- 3) Pelarut atau air, yang mengatur viskositas dan membantu proses aplikasi.
- 4) Aditif, digunakan untuk mengatur sifat tertentu, seperti waktu pengeringan, kestabilan, aliran, ketahanan terhadap cuaca, dan tingkat kilap.



Gambar 2.3 Komponen utama cat

Warna cat terbentuk melalui penyerapan dan pemantulan cahaya oleh

pigmen. Pencampuran pigmen mengikuti mekanisme substraktif karena setiap pigmen menyerap bagian tertentu dari spektrum cahaya dan memantulkan bagian lainnya (Eiseman, 2024). Kombinasi beberapa pigmen kemudian menghasilkan warna dengan karakteristik tertentu.

Setelah diaplikasikan, cat mengalami pembentukan lapisan melalui penguapan medium cair, pemadatan, dan perubahan susunan partikel. Proses tersebut dapat menyebabkan warna cat setelah mengering berbeda dari kondisi basah. Tampilan akhir cat juga dapat dipengaruhi oleh formulasi, ketebalan lapisan, karakteristik permukaan, pencahayaan, dan kondisi lingkungan selama pengeringan. Oleh karena itu, penelitian ini menggunakan warna cat kering sebagai objek pengujian karena merepresentasikan kondisi cat setelah diaplikasikan pada permukaan.

2.1.4 *Sistem Tinting*

Sistem *tinting* merupakan teknologi pencampuran warna berbasis komputer yang menghasilkan berbagai pilihan warna melalui pencampuran *base paint* dan tinta dasar berdasarkan formula dalam basis data (Ekstrand, 2023). Sistem ini umumnya terdiri atas mesin dispenser, perangkat lunak pengelola formula, dan alat pencampur. Ketika warna dipilih dari katalog, perangkat lunak mengambil formula yang berisi jenis dan takaran tinta, kemudian mesin dispenser menuangkan tinta ke dalam *base paint* untuk dicampur secara merata. Penggunaan formula digital dapat mempercepat penyediaan warna, mengurangi kebutuhan penyimpanan cat siap pakai, dan membantu menjaga konsistensi hasil pencampuran. Namun, warna akhir tetap dapat dipengaruhi oleh kalibrasi mesin, kondisi bahan, ketelitian penakaran, proses pencampuran, dan kondisi aplikasi cat.

Sistem *tinting* yang menjadi objek penelitian ini menggunakan formula yang telah tersimpan dalam basis data, tetapi belum menyediakan fungsi *reverse tinting*. Dalam penelitian ini, *reverse tinting* merujuk pada proses memperkirakan jenis tinta dasar berdasarkan warna cat yang telah terbentuk atau mengering. Proses tersebut kompleks karena hubungan antara tampilan warna dan formula pembentuknya tidak selalu bersifat satu-satu. Kombinasi tinta yang berbeda dapat menghasilkan warna serupa, sedangkan tampilan cat kering dapat dipengaruhi oleh *base paint*, ketebalan lapisan, permukaan, pencahayaan, dan proses pengeringan. Penelitian ini

membatasi proses *reverse tinting* pada prediksi keberadaan jenis tinta dasar tanpa memperkirakan takarannya.

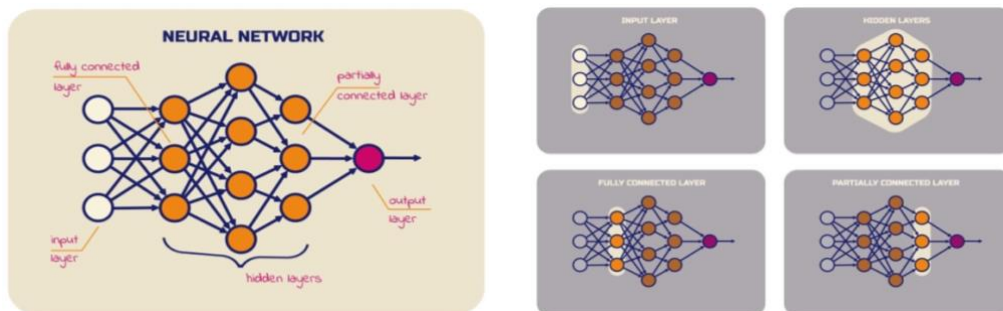


Gambar 2.4 Komponen sistem *tinting*

2.1.5 Deep Learning

Deep learning merupakan bagian dari *machine learning* yang menggunakan jaringan saraf tiruan dengan banyak lapisan untuk mempelajari representasi fitur secara bertingkat langsung dari data. Kemampuan tersebut mengurangi kebutuhan terhadap ekstraksi fitur manual dan memungkinkan model mempelajari pola sederhana hingga fitur yang lebih kompleks. Oleh karena itu, *deep learning* banyak digunakan untuk mengolah data berdimensi tinggi, seperti citra, suara, dan teks (Goodfellow dkk., 2016).

Jaringan saraf pada *deep learning* terdiri atas lapisan *input*, lapisan tersembunyi, dan lapisan *output*. Setiap neuron menghitung kombinasi antara nilai masukan dan bobot, kemudian menerapkan fungsi aktivasi non-linear. Susunan lapisan tersebut memungkinkan jaringan mempelajari representasi data secara bertahap. Semakin dalam jaringan, semakin kompleks representasi yang dapat dipelajari, meskipun peningkatan kedalaman juga dapat meningkatkan kebutuhan komputasi dan risiko *overfitting* (LeCun dkk., 2015).



Gambar 2.5 Struktur dasar jaringan saraf tiruan

Proses pelatihan dimulai dengan *forward propagation* untuk menghasilkan

prediksi dan menghitung nilai fungsi *loss*. Selanjutnya, *backpropagation* digunakan untuk menghitung gradien terhadap parameter jaringan. Gradien tersebut dimanfaatkan oleh algoritma optimasi, seperti SGD, Adam, atau AdamW, untuk memperbarui bobot sehingga nilai *loss* dapat diminimalkan.

Fungsi aktivasi memberikan sifat non-linear agar jaringan dapat mempelajari hubungan yang kompleks. Dalam penelitian ini, ReLU digunakan pada lapisan tersembunyi karena memiliki perhitungan yang sederhana dan membantu mengurangi masalah *vanishing gradient*. Sementara itu, fungsi *sigmoid* digunakan pada lapisan *output* untuk menghasilkan probabilitas setiap label secara terpisah pada *multi-label classification* (Ramachandran dkk., 2017).

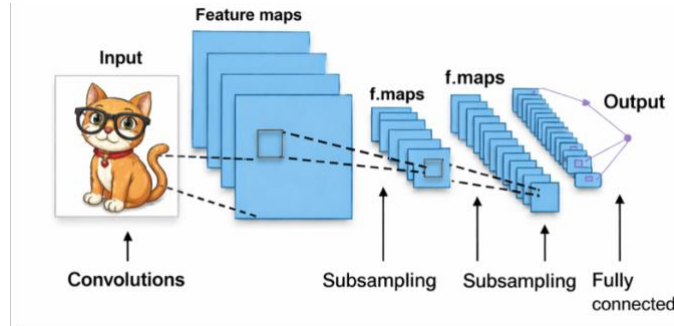
Tabel 2.1 Fungsi Aktivasi pada *Deep Learning*

Fungsi	Rumus	Rentang Keluaran	Penggunaan Umum
Sigmoid	$\frac{1}{1 + e^{-x}}$	0 – 1	Lapisan <i>output</i> klasifikasi biner dan multi-label
Softmax	$\frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$	0 – 1 dengan jumlah probabilitas 1	Lapisan <i>output</i> klasifikasi <i>multi-class</i> yang saling eksklusif
ReLU	$\max(0, x)$	0 hingga tak terhingga	<i>Hidden layer</i> jaringan saraf

Untuk pengolahan citra, salah satu arsitektur *deep learning* yang banyak digunakan adalah *Convolutional Neural Network* (CNN). Model berkapasitas besar yang dilatih menggunakan data terbatas berisiko mengalami *overfitting*, yaitu kondisi ketika model memiliki performa baik pada data pelatihan, tetapi tidak mampu melakukan generalisasi dengan baik pada data baru (Belkin dkk., 2019). Risiko tersebut dapat dikurangi melalui teknik seperti *dropout*, *weight decay*, *early stopping*, dan *transfer learning*.

2.1.6 *Convolutional Neural Network*

Convolutional Neural Network (CNN) merupakan jenis jaringan saraf yang dirancang untuk mempelajari pola spasial dalam data visual, seperti citra dan video (Goodfellow dkk., 2016). CNN menggunakan operasi konvolusi untuk mengekstraksi fitur secara bertingkat. Lapisan awal umumnya mempelajari fitur sederhana, seperti tepi, perubahan intensitas, dan gradien warna, sedangkan lapisan dalam membentuk representasi yang lebih kompleks, seperti tekstur, pola warna, dan bentuk.



Gambar 2.6 Struktur umum *Convolutional Neural Network*

Lapisan konvolusi menggunakan sejumlah *kernel* yang digeser pada citra untuk mendeteksi pola lokal. Hasil operasi tersebut membentuk *feature map* yang menunjukkan respons terhadap fitur tertentu pada setiap posisi spasial (He dkk., 2016). Penggunaan *kernel* yang sama pada berbagai posisi memungkinkan CNN mendeteksi fitur serupa pada lokasi yang berbeda. Meskipun demikian, ketahanan terhadap perubahan orientasi dan skala tetap dipengaruhi oleh arsitektur serta karakteristik data pelatihan (Martins dkk., 2025). Persamaan 2-1 menunjukkan bentuk dasar operasi konvolusi dengan asumsi *stride* satu. Pada citra RGB, jumlah kanal masukan adalah tiga, yaitu merah, hijau, dan biru.

$$Z_{i,j,k} = \sum_{m=0}^{f-1} \sum_{n=0}^{f-1} \sum_{c=0}^{C-1} X_{i+m,j+n,c} \cdot K_{m,n,c,k} + b_k \quad 2-1$$

Di mana

$Z_{i,j,k}$: Nilai *feature map* pada posisi (i, j) untuk filter ke-k

X : Citra *input*

K : Kernel/*filter*

b_k : Bias *filter* ke-k

f : Ukuran kernel

i, j : Posisi spasial pada *feature map*

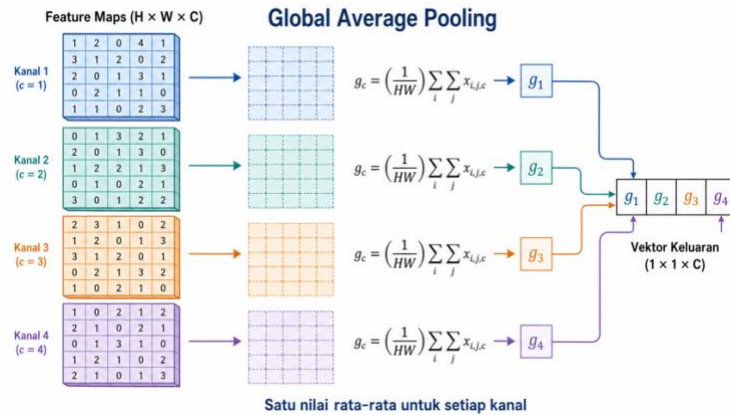
m, n : Posisi elemen dalam kernel

c : Indeks kanal citra masukan

C : Jumlah kanal citra masukan

Ukuran spasial *feature map* dapat dikurangi menggunakan lapisan *pooling*. Pada penelitian ini, hasil ekstraksi fitur diringkas menggunakan *Global Average Pooling* (GAP), yaitu dengan menghitung nilai rata-rata setiap kanal pada seluruh posisi spasial. GAP mengubah *feature map* menjadi vektor fitur dan dapat mengurangi jumlah parameter dibandingkan penggunaan *flatten* yang langsung

dihubungkan dengan lapisan padat (Goodfellow dkk., 2016).



Gambar 2.7 Ilustrasi proses *Global Average Pooling* pada setiap kanal *feature map*

$$g_k = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W Z_{i,j,k} \quad 2-2$$

Di mana

g_k : Nilai rata-rata global kanal ke-k

$Z_{i,j,k}$: Nilai *feature map* pada posisi (i, j) dan kanal ke-k

H : Tinggi *feature map*

W : Lebar *feature map*

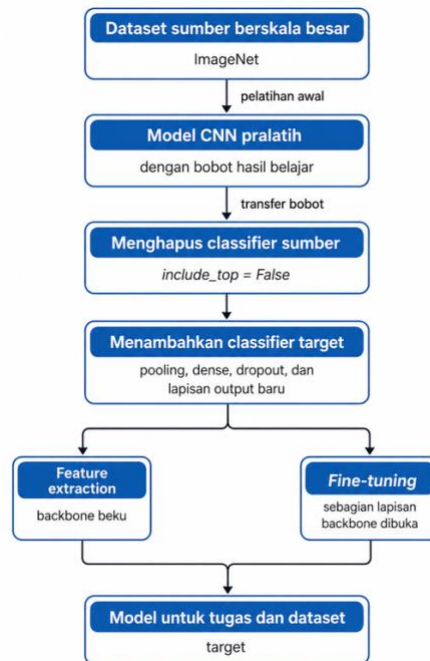
Vektor yang dihasilkan GAP kemudian diteruskan ke *classification head* untuk menghasilkan prediksi. Dalam penerapan *transfer learning*, bagian konvolusi model CNN pralatih digunakan sebagai *backbone*, sedangkan lapisan klasifikasi aslinya diganti dengan lapisan klasifikasi yang sesuai dengan tugas penelitian. Parameter *backbone* selanjutnya dapat dibekukan atau disesuaikan melalui proses *fine-tuning*.

2.1.7 Transfer Learning

Transfer learning merupakan pendekatan yang memanfaatkan pengetahuan dari model yang telah dilatih pada tugas dan data sumber untuk membantu pembelajaran tugas target. Pengetahuan tersebut umumnya berupa bobot dan representasi fitur yang telah dipelajari sebelumnya. Pendekatan ini sesuai untuk kondisi ketika jumlah data target terbatas karena model tidak harus memulai proses pembelajaran dari bobot acak (Zhuang dkk., 2020).

Dalam penelitian ini, model CNN dengan bobot pralatih ImageNet digunakan untuk mengembangkan model prediksi kombinasi tinta dasar. ImageNet menjadi

data sumber yang digunakan dalam pembentukan bobot awal, sedangkan 488 citra digital warna katalog menjadi data target. Meskipun tugas sumber berupa klasifikasi objek berbeda dengan tugas target berupa *multi-label classification*, model pralatih telah mempelajari fitur visual umum, seperti tepi, perubahan intensitas, kontras, tekstur, dan bentuk, yang dapat digunakan sebagai representasi awal (Russakovsky dkk., 2015).



Gambar 2.8 Alur Umum *Transfer Learning*

Model *transfer learning* untuk klasifikasi citra umumnya terdiri atas *backbone* dan *classification head*. *Backbone* merupakan bagian CNN pralatih yang berfungsi sebagai ekstraksi fitur, sedangkan *classification head* mengubah representasi fitur menjadi hasil prediksi sesuai tugas target. Lapisan klasifikasi asli model dihilangkan karena keluaran ImageNet tidak sesuai dengan sembilan label tinta yang diprediksi dalam penelitian (Kornblith dkk., 2019).

Penerapan *transfer learning* dapat dilakukan melalui strategi *feature extraction* atau *fine-tuning*. Pada *feature extraction*, seluruh parameter *backbone* dibekukan dan pelatihan hanya dilakukan terhadap *classification head*. Pada *fine-tuning*, sebagian lapisan *backbone* dibuka agar bobotnya dapat disesuaikan dengan karakteristik data target. Lapisan akhir biasanya lebih dahulu dipilih untuk dibuka karena cenderung mempelajari fitur yang lebih spesifik terhadap tugas sumber,

sedangkan lapisan awal mempelajari fitur visual yang lebih umum (Yosinski dkk., 2014).

Fine-tuning dapat dilakukan secara bertahap atau langsung. Pada strategi bertahap, *classification head* dilatih terlebih dahulu ketika *backbone* masih dibekukan, kemudian sebagian lapisan *backbone* dibuka. Penelitian ini menggunakan *one-stage fine-tuning*, yaitu melatih *classification head* dan sejumlah lapisan akhir *backbone* secara bersamaan sejak awal. Jumlah lapisan yang dibuka dan konfigurasi pelatihannya ditentukan melalui beberapa percobaan. Strategi ini memerlukan pengaturan *learning rate* yang sesuai agar pembaruan bobot tidak menghilangkan representasi awal yang telah dipelajari model pralatih.

Keras Applications menyediakan berbagai arsitektur CNN dengan bobot pralatih ImageNet. Beberapa arsitektur CNN pralatih yang dapat digunakan dalam *transfer learning* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Perbandingan Beberapa Arsitektur CNN Pralatih

Arsitektur	Konsep Utama	Parameter	Top-1 ImageNet	Kelebihan
ResNet50	Residual learning dan shortcut connection	25,6 juta	74,9%	Membantu mengatasi kesulitan optimasi jaringan dalam. Aliran gradien lebih stabil. Banyak digunakan sebagai backbone transfer learning.
EfficientNetB2	Compound scaling dan MBConv	9,2 juta	80,1%	Menyeimbangkan kedalaman, lebar, dan resolusi jaringan.
DenseNet121	Dense connectivity dan penggunaan kembali fitur	8,1 juta	75,0%	Memperkuat propagasi fitur dan gradien. Mengurangi masalah vanishing gradient. Mendukung penggunaan kembali fitur dari berbagai kedalaman. Jumlah parameternya relatif efisien.
MobileNetV2	Inverted residual dan linear bottleneck	3,5 juta	71,3%	Jumlah parameter dan ukuran model paling kecil. Waktu inferensi relatif cepat. Sesuai untuk aplikasi web dan perangkat dengan sumber daya terbatas.

Arsitektur	Konsep Utama	Parameter	Top-1 ImageNet	Kelebihan
NASNetMobile	Neural Architecture Search	5,3 juta	74,4%	Arsitektur diperoleh melalui pencarian otomatis. Jumlah parameter relatif rendah. Dirancang untuk keseimbangan akurasi dan komputasi pada platform bergerak.

Pemanfaatan bobot pralatih dapat mempercepat proses pembelajaran dan mengurangi kebutuhan terhadap data target dalam jumlah besar. Namun, model dengan performa tinggi pada ImageNet belum tentu menghasilkan performa terbaik pada citra warna cat. Keberhasilan proses transfer tetap dipengaruhi oleh perbedaan domain, karakteristik arsitektur, jumlah lapisan yang dibuka, dan konfigurasi pelatihan (Zhuang dkk., 2020). Oleh karena itu, arsitektur terbaik dalam penelitian ini ditentukan berdasarkan hasil evaluasi pada data target, bukan berdasarkan akurasi ImageNet.

2.1.8 Multi-Label Classification

Multi-label classification merupakan pendekatan klasifikasi di mana satu data dapat memiliki lebih dari satu label secara bersamaan (Tsoumakas & Katakis, 2007). Pendekatan ini berguna dalam banyak kasus seperti diagnosis beberapa penyakit, pengenalan emosi, dan formulasi warna cat berbasis kombinasi tinta dasar. Berbeda dari klasifikasi tunggal, evaluasi model multi-label memerlukan beberapa metrik untuk menilai performa dari berbagai aspek. Zhang dan Zhou (2014) menjelaskan bahwa kombinasi antara *Hamming Loss*, *Precision*, *Recall*, *F1-Score*, dan *Subset Accuracy* memberikan pandangan menyeluruh terhadap kemampuan model dalam mengenali pola label majemuk.

1) *Hamming Loss* (HL)

Hamming Loss merupakan metrik yang paling mendasar untuk mengevaluasi model multi-label.

$$HL = \frac{1}{n \times |L|} \sum_{i=1}^n |Y_i \Delta \hat{Y}_i| \quad 2-3$$

di mana

- n : Jumlah sampel
- $|L|$: Jumlah total label
- Y_i : Label sebenarnya
- $\hat{Y}_i$: label hasil prediksi

Hamming Loss menghitung proporsi kesalahan klasifikasi pada seluruh label dan sampel. Semakin kecil nilainya, semakin baik performa model. Keunggulan metrik ini adalah kemampuannya mendeteksi kesalahan kecil pada label individu tanpa mengharuskan seluruh kombinasi label benar. Namun, kelemahannya adalah tidak mempertimbangkan hubungan antar label. Model bisa memiliki nilai *Hamming Loss* rendah tetapi gagal memahami korelasi antar kombinasi tinta.

2) *Precision* (P)

Precision mengukur sejauh mana label yang diprediksi positif oleh model benar-benar relevan.

$$P = \frac{TP}{TP + FP} \quad 2-4$$

di mana

TP (True Positive) : Jumlah label yang benar-benar relevan dan berhasil diprediksi

FP (False Positive) : Label yang salah diprediksi positif

Nilai *precision* tinggi menunjukkan bahwa model jarang salah mendeteksi label yang tidak seharusnya ada. Dalam konteks warna cat, *precision* tinggi berarti model jarang menambahkan tinta yang tidak relevan pada campuran warna. *Precision* lebih penting dalam aplikasi di mana kesalahan menambahkan label berakibat fatal, misalnya pada diagnosis penyakit atau formulasi kimia, karena prediksi berlebih dapat menghasilkan hasil yang salah total.

3) *Recall* (R)

Recall mengukur kemampuan model dalam mendeteksi semua label yang seharusnya ada.

$$R = \frac{TP}{TP + FN} \quad 2-5$$

di mana

TP (True Positive) : Jumlah label yang benar-benar relevan dan berhasil diprediksi

FN (False Negative) : Label yang seharusnya muncul tetapi tidak terdeteksi oleh model

Nilai *recall* yang tinggi menunjukkan bahwa model jarang melewatkan label penting. Pada sistem prediksi warna cat, *recall* tinggi berarti model mampu mengenali semua tinta dasar yang benar-benar diperlukan untuk menghasilkan warna target. Namun, *recall* tinggi tanpa *precision* yang memadai dapat menghasilkan banyak kesalahan tambahan. Oleh karena itu, *recall* sebaiknya dievaluasi bersama *precision* agar hasil lebih seimbang.

4) *F1-Score* (F1)

F1-Score adalah harmonisasi antara *precision* dan *recall*, yang mengukur keseimbangan antara dua aspek tersebut.

$$F1 = 2 \times \frac{P \times R}{P + R} \quad 2-6$$

di mana

P (*Precision*) : Rasio antara jumlah label yang benar-benar relevan dan berhasil diprediksi benar dibandingkan dengan seluruh label yang diprediksi positif oleh model

R (*Recall*) : Rasio antara jumlah label yang benar-benar relevan dan berhasil dibandingkan dengan seluruh label yang seharusnya relevan

F1-score memberikan nilai yang tinggi hanya jika *precision* dan *recall* sama-sama tinggi, sehingga model yang memiliki salah satu nilai rendah tidak akan terlihat baik. Dalam kasus prediksi kombinasi warna, *F1-score* dapat membantu menilai seberapa baik model menyeimbangkan kemampuan untuk menemukan semua tinta relevan dengan akurasi dalam memilih tinta yang tepat.

5) *Subset Accuracy* (SA)

Subset Accuracy merupakan metrik paling ketat dalam evaluasi multi-label karena hanya memberikan nilai satu jika seluruh label prediksi tepat sama dengan label aktual.

$$SA = \frac{1}{n} \sum_{i=1}^n I(Y_i = \hat{Y}_i) \quad 2-7$$

di mana

I : Fungsi indikator yang bernilai 1 jika seluruh label cocok dan 0 jika tidak

Y_i : Label sebenarnya

$\hat{Y}_i$: Label hasil prediksi

Metrik ini memberikan pandangan yang jelas terhadap kesempurnaan prediksi, tetapi nilainya sering kali rendah terutama pada dataset dengan banyak label. Dalam konteks industri cat, *subset accuracy* dapat menggambarkan seberapa sering sistem AI mampu memprediksi komposisi tinta yang benar-benar identik dengan formula sebenarnya. Namun, karena sensitivitasnya sangat tinggi terhadap satu kesalahan kecil, metrik ini biasanya digunakan bersamaan dengan *Hamming Loss* untuk mendapatkan evaluasi yang lebih realistis.

2.1.9 Computer Vision dan Karakteristik Citra Digital

Computer vision merupakan bidang dari kecerdasan buatan yang mempelajari metode untuk memperoleh dan menganalisis informasi dari citra atau video digital (Szeliski, 2022). Dalam penelitian ini, *computer vision* digunakan untuk mengolah citra warna cat sebagai masukan model prediksi kombinasi tinta dasar. Pendekatan berbasis *deep learning* memungkinkan CNN mempelajari representasi visual langsung dari nilai piksel tanpa memerlukan ekstraksi fitur secara manual.

Citra digital tersusun atas piksel yang memiliki nilai intensitas pada posisi tertentu. Pada citra RGB, setiap piksel direpresentasikan melalui kanal merah, hijau, dan biru. Citra dengan kedalaman 8-bit per kanal memiliki nilai piksel antara 0 hingga 255 sehingga dapat direpresentasikan sebagai tensor dengan dimensi tinggi, lebar, dan tiga kanal warna (Gonzalez & Woods, 2018). Sebelum dimasukkan ke dalam model, ukuran citra dan skala nilai piksel disesuaikan dengan kebutuhan arsitektur CNN. Tahapan penyesuaian tersebut dijelaskan lebih lanjut pada metodologi penelitian.



Gambar 2.9 Perbedaan karakteristik citra katalog dan foto sampel cat fisik

Citra katalog dan foto cat fisik memiliki karakteristik yang berbeda. Citra katalog merupakan data digital dengan warna dan latar yang relatif seragam,

sedangkan foto cat fisik dipengaruhi oleh pencahayaan, sensor kamera, eksposur, *white balance*, jarak dan sudut pengambilan, tekstur permukaan, serta refleksi cahaya. Kamera juga melakukan pemrosesan internal, seperti koreksi warna dan transformasi non-linear, sebelum menghasilkan citra RGB. Oleh karena itu, nilai warna pada foto tidak hanya ditentukan oleh warna fisik objek, tetapi juga oleh kondisi pengambilan dan pemrosesan kamera (Afifi dkk., 2019; Karaimer & Brown, 2018).

Pengendalian perangkat kamera, posisi pengambilan, dan pencahayaan dapat mengurangi variasi selama akuisisi foto, tetapi tidak sepenuhnya menyamakan karakteristik foto dengan citra katalog. Perbedaan distribusi antara data pengembangan dan data penerapan tersebut disebut *domain shift*. Kondisi ini dapat menurunkan kemampuan generalisasi model ketika data pengujian memiliki karakteristik yang berbeda dari data pelatihan (Recht dkk., 2019; Torralba & Efros, 2011).

Augmentasi dapat digunakan untuk menghasilkan variasi transformasi dari citra pelatihan, misalnya melalui perubahan kecerahan, kontras, atau karakteristik warna (Cubuk dkk., 2019). Transformasi harus disesuaikan dengan informasi yang menjadi dasar prediksi agar tidak mengubah makna label. Meskipun augmentasi dapat meningkatkan keragaman data, teknik tersebut tidak menjamin seluruh perbedaan antara citra katalog dan foto cat fisik dapat diatasi.

2.2 Penelitian Terdahulu

Penelitian mengenai *multi-label image classification* berkembang untuk menangani citra yang memiliki lebih dari satu label secara bersamaan. Berbagai pendekatan telah digunakan, mulai dari CNN konvensional dan *transfer learning* hingga GCN dan Transformer. Pendekatan tersebut berbeda dalam cara mengekstraksi fitur, menangani ketidakseimbangan data, dan memodelkan hubungan antarlabel.

Bariyah & Arif Rasyidi (2021) menerapkan CNN untuk mengidentifikasi beberapa motif yang terdapat dalam satu citra batik. Hasil penelitian menunjukkan bahwa pendekatan *multi-label* dapat merepresentasikan keberadaan beberapa motif secara bersamaan dengan akurasi yang dilaporkan lebih dari 91%. Namun, model yang digunakan masih berupa CNN dasar dan belum memanfaatkan model prelatih

maupun memodelkan hubungan antarmotif secara eksplisit. Penelitian tersebut menunjukkan penerapan *multi-label classification* pada konteks citra lokal, tetapi masih membuka peluang penggunaan arsitektur pralatih yang lebih kuat (Bariyah & Arif Rasyidi, 2021).

Penerapan *multi-label classification* juga banyak dikembangkan pada citra medis. Wang dkk. (2017) menggunakan CNN untuk mendeteksi beberapa penyakit pada dataset ChestX-ray8 yang terdiri atas 108.948 citra. Penelitian tersebut menunjukkan bahwa satu model dapat menghasilkan beberapa label penyakit, tetapi penggunaan label yang diperoleh secara otomatis dari laporan radiologi berpotensi menimbulkan kesalahan pelabelan (Wang dkk., 2017). Baltruschat dkk. (2019) kemudian membandingkan pelatihan dari awal, ekstraksi fitur, dan *fine-tuning* menggunakan beberapa varian ResNet pada citra X-ray. Hasilnya menunjukkan bahwa strategi pelatihan dan penyesuaian lapisan dapat memberikan perbedaan performa yang cukup besar (Baltruschat dkk., 2019). Sementara itu, Ge dkk. (2018) menggunakan DenseNet dan ResNet pralatih serta fungsi *loss* yang dirancang untuk menangani *multi-label classification* dan ketidakseimbangan data. Ketiga penelitian tersebut mendukung penggunaan CNN dan *transfer learning*, tetapi hasilnya tidak dapat diterapkan secara langsung pada citra warna cat karena perbedaan karakteristik domain dan data.

Kornblith dkk. (2019) membandingkan 16 arsitektur pada 12 dataset target untuk mengevaluasi hubungan antara performa model pada ImageNet dan kemampuan transfernya. Hasil penelitian menunjukkan bahwa performa ImageNet berkorelasi kuat dengan performa pada sebagian besar tugas target. Namun, manfaat prapelatihan tidak selalu sama pada semua dataset dan beberapa teknik yang meningkatkan akurasi ImageNet tidak selalu menghasilkan representasi transfer yang lebih baik. Temuan tersebut menunjukkan bahwa arsitektur dengan akurasi ImageNet tinggi belum tentu menjadi pilihan terbaik untuk data target sehingga tetap perlu dievaluasi secara empiris (Kornblith dkk., 2019).

Selain ekstraksi fitur visual, beberapa penelitian berusaha memodelkan hubungan antarlabel. Z.-M. Chen dkk. (2019) mengembangkan ML-GCN yang merepresentasikan label sebagai graf dan menggunakan GCN untuk mempelajari hubungan antarkategori. Model tersebut memberikan hasil lebih baik daripada

sejumlah metode pembandingan pada MS-COCO dan PASCAL VOC 2007. Meskipun demikian, pemrosesan fitur visual dan hubungan semantik label masih dilakukan melalui komponen yang berbeda sehingga meningkatkan kompleksitas model (Z.-M. Chen dkk., 2019).

Liu dkk. (2021) mengembangkan Query2Label dengan menggunakan label *embedding* sebagai *query* pada *Transformer decoder*. Pendekatan tersebut memungkinkan model mengambil fitur visual yang relevan untuk setiap label dan menghasilkan performa yang kompetitif pada beberapa dataset citra *multi-label*. Namun, penggunaan Transformer meningkatkan kompleksitas pelatihan dan kebutuhan komputasi, terutama jika dibandingkan dengan CNN yang lebih ringan. Selain itu, evaluasinya dilakukan pada dataset citra objek berskala besar sehingga penerapannya pada dataset kecil masih memerlukan pengujian lebih lanjut (Liu dkk., 2021).

Ketidakseimbangan antara label positif dan negatif dibahas oleh Ben-Baruch dkk. (2021) melalui *Asymmetric Loss*. Fungsi loss tersebut mengurangi pengaruh label negatif yang mudah agar proses optimasi tidak didominasi oleh jumlah label negatif yang jauh lebih besar. Metode ini memberikan performa kompetitif pada sejumlah dataset *multi-label*, tetapi efektivitasnya dipengaruhi oleh konfigurasi parameter asimetris. Penelitian tersebut memperlihatkan pentingnya pemilihan fungsi *loss* ketika distribusi label tidak seimbang (Ben-Baruch dkk., 2021).

Berdasarkan tinjauan pustaka yang telah dibahas, dapat disimpulkan bahwa *multi-label image classification* telah berkembang dari penggunaan CNN konvensional hingga pendekatan *transfer learning*, GCN, dan Transformer. Meskipun GCN dan Transformer mampu memodelkan hubungan antarlabel lebih eksplisit, pendekatan *transfer learning* berbasis CNN tetap relevan untuk dataset berukuran terbatas karena dapat memanfaatkan fitur visual dari model prelatih, mengurangi kebutuhan data besar, dan menekan risiko *overfitting*. Perbedaan performa antar arsitektur serta permasalahan ketidakseimbangan label menunjukkan bahwa pemilihan *backbone*, *fine-tuning*, dan fungsi *loss* perlu diuji lebih lanjut.

Tabel 2.3 Perbandingan Model Penelitian Terdahulu

No	Penulis	Model	Keterkaitan
1	Bariyah & Arif Rasyidi (2021)	CNN multi-label classification motif batik	- Relevan pada kasus satu citra dengan lebih dari satu label. - Menguatkan penggunaan CNN pada domain visual spesifik. - Belum memodelkan hubungan antar label secara eksplisit.
2	Wang dkk. (2017)	CNN multi-label dan <i>weakly supervised localization</i> pada ChestX-ray8	- Menjadi dasar penerapan <i>multi-label classification</i> berbasis citra. - Satu model dapat menghasilkan beberapa label secara bersamaan. - Menggunakan dataset yang lebih besar.
3	Baltruschat dkk. (2019)	ResNet dengan <i>training from scratch</i> , <i>transfer learning</i> , dan <i>fine-tuning</i>	- Membandingkan pelatihan dari awal dengan <i>transfer learning</i>. - Membandingkan <i>transfer learning</i> tanpa dan dengan <i>fine-tuning</i>. - Mendukung penggunaan model pralatih pada <i>multi-label classification</i>.
4	Kornblith dkk. (2019)	Enam belas arsitektur pralatih ImageNet dengan <i>fixed feature extraction</i> dan <i>fine-tuning</i>	- Membandingkan kemampuan <i>transfer</i> dari berbagai arsitektur pralatih. - Menunjukkan hubungan antara performa pada ImageNet dan performa pada tugas target. - Mendukung perbandingan beberapa arsitektur <i>transfer learning</i>. - Menguatkan evaluasi yang mendalam dalam pemilihan <i>backbone</i>.
5	Ge dkk. (2018)	DenseNet dan ResNet berbasis <i>transfer learning</i> dengan <i>multi-label softmax loss</i>	- Menerapkan <i>transfer learning</i> pada <i>multi-label image classification</i>. - Menggunakan DenseNet dan ResNet sebagai ekstraktor fitur. - Membahas kemiripan visual antarkelas dan ketidakseimbangan label. - Menjadi pembanding bagi penggunaan DenseNet121

No	Penulis	Model	Keterkaitan
			dan <i>weighted binary cross-entropy</i> .
6	Z.-M. Chen dkk. (2019)	Multi-Label Graph Convolutional Network (ML-GCN)	- Menggabungkan CNN dengan Graph Convolutional Network. - Memodelkan korelasi dan kemunculan bersama antarlabel. - Menunjukkan bahwa label tidak selalu bersifat independen. - Menjadi pembanding bagi keluaran <i>sigmoid</i> independen.
7	Liu dkk. (2021)	Query2Label menggunakan <i>vision backbone</i> dan Transformer decoder	- Menggunakan setiap label sebagai query untuk memperoleh fitur visual yang relevan. - Memanfaatkan Transformer untuk mempelajari informasi spesifik setiap label. - Menjadi pembanding metode <i>multi-label classification</i> modern. - Memiliki kompleksitas lebih tinggi dibandingkan <i>transfer learning</i> berbasis CNN.
8	Ben-Baruch dkk. (2021)	Model <i>multi-label classification</i> dengan <i>Asymmetric Loss</i>	- Membahas ketidakseimbangan antara label positif dan negatif. - Mengurangi kontribusi label negatif yang mudah selama pelatihan. - Menunjukkan pentingnya pemilihan fungsi <i>loss</i> pada data multi-label. - Menjadi pembanding bagi penggunaan <i>weighted binary cross-entropy</i>.