

# Identifikasi Jenis Ikan Cupang Menggunakan Metode Vision Transformer Dan Deep Learning

Nuke Lu'luul Chusna<sup>1</sup>, Harry Dwiyan K<sup>2</sup>, Ahmad Daffa Muwafaqi<sup>3</sup>, Imam Muttaqin<sup>4</sup>, Asep Mulyana<sup>5</sup>  
Universitas Krisnadwipayana<sup>1</sup>, Universitas Krisnadwipayana<sup>2</sup>, Universitas Krisnadwipayana<sup>3</sup>, Universitas Krisnadwipayana<sup>4</sup>, Universitas Krisnadwipayana<sup>5</sup>  
email : [nukelchusna@unkris.ac.id](mailto:nukelchusna@unkris.ac.id), [harrydwiyan@gmail.com](mailto:harrydwiyan@gmail.com), [ahdaffa24@gmail.com](mailto:ahdaffa24@gmail.com), [imam.muttaqin@unkris.ac.id](mailto:imam.muttaqin@unkris.ac.id), [asep.m@unkris.ac.id](mailto:asep.m@unkris.ac.id)

(Naskah masuk: 10 April 2026; diterima untuk diterbitkan: 5 Mei 2026)

**Abstrak** — Penelitian ini bertujuan untuk mengidentifikasi dan mengklasifikasikan jenis ikan cupang menggunakan metode *Vision Transformer (ViT)* dalam pemrosesan citra. Model *ViT* dipilih karena kemampuannya dalam menangkap representasi spasial global secara efektif dibandingkan arsitektur konvensional berbasis *CNN*. Dataset yang digunakan berasal dari dua sumber, yaitu situs *Kaggle* dan hasil pencarian manual melalui *Google Images*, yang mencakup tiga jenis ikan cupang, yaitu *Halfmoon*, *Plakat*, dan *Crowntail*. Sistem dikembangkan menggunakan *PyTorch* dan *Google Colab*, dengan beberapa skenario pembagian data latih dan data uji, yaitu 80:20, 70:30, 60:40, dan 50:50. Hasil pengujian menunjukkan bahwa model mencapai performa terbaik pada rasio 80:20, sementara pada rasio 50:50 dan 60:40 mulai terlihat penurunan akurasi sebesar 0,01. Secara keseluruhan, metode *Vision Transformer* terbukti efektif dalam melakukan klasifikasi jenis ikan cupang berdasarkan citra, meskipun performanya sensitif terhadap proporsi data latih yang tersedia.

**Kata Kunci** : *Vision Transformer*, klasifikasi citra, ikan cupang, *machine learning*, identifikasi ikan.

## 1. PENDAHULUAN

Ikan cupang (*Betta splendens*) telah dikenal luas sebagai salah satu ikan hias air tawar yang paling diminati di seluruh dunia, khususnya di Indonesia. Ikan ini terkenal karena keindahan *Visualnya* yang memukau, dengan warna-warna cerah, bentuk sirip yang unik, dan pola tubuh yang sangat beragam. Keunikan tersebut menjadikannya pilihan utama bagi banyak penghobi yang mencari peliharaan dengan daya tarik estetika tinggi. Selain itu, ikan cupang juga dapat dipelihara di ruang terbatas, menjadikannya ideal untuk penghobi yang tinggal di daerah dengan keterbatasan ruang, seperti apartemen atau rumah dengan halaman kecil. Popularitas ikan cupang sebagai ikan hias terus meningkat seiring dengan semakin tingginya minat masyarakat terhadap gaya hidup yang berfokus pada estetika dan keindahan (Arshal, 2023; Rinata et al., 2023).

Keanekaragaman jenis ikan cupang sangat luas, dengan estimasi mencapai sekitar 73 jenis yang berbeda. Di Indonesia, terdapat lebih dari 13 kompleks ikan cupang, dengan jenis-jenis seperti *Halfmoon*, *Plakat*, *Spadetail*, *MultiColored*, dan *Butterfly* yang paling dikenal dan populer di kalangan penghobi. Keanekaragaman ini menjadi daya tarik utama bagi banyak penghobi, yang seringkali mencari ikan dengan variasi *visual* yang menarik. Namun, hal ini juga menciptakan tantangan dalam hal identifikasi, terutama bagi penghobi baru atau mereka yang belum berpengalaman dalam mengenali jenis ikan cupang.

Proses identifikasi jenis ikan cupang memerlukan keahlian

khusus dan pengalaman yang cukup mendalam, serta memakan waktu yang lama. Dalam hal ini, kesulitan identifikasi tidak hanya berdampak pada penghobi yang ingin mengenali koleksi mereka, tetapi juga pada pelaku bisnis dan pembudidaya yang memerlukan identifikasi cepat dan akurat untuk tujuan klasifikasi, penetapan harga, dan pemasaran. Kesalahan dalam identifikasi dapat menurunkan nilai jual ikan atau menyebabkan kerugian ekonomi yang cukup besar. Hal ini menunjukkan perlunya sistem identifikasi yang lebih efisien dan akurat, terutama untuk mendukung skala komersial di mana kecepatan dan akurasi menjadi faktor penting dalam operasional bisnis (Hasym & Susilawati, 2021).

Salah satu metode yang sering digunakan adalah *Histogram of Oriented Gradient (HOG)*, yang memiliki keunggulan dalam ketahanan terhadap variasi pencahayaan pada citra. Namun, metode ini kurang efektif dalam mengekstraksi fitur pada citra dengan tingkat kompleksitas tinggi, seperti gambar ikan cupang dengan berbagai variasi warna dan bentuk yang sangat halus.

*Deep learning*, khususnya dengan menggunakan *Convolutional Neural Networks (CNN)*, telah menunjukkan kemampuan luar biasa dalam berbagai aplikasi pengolahan citra, termasuk dalam klasifikasi objek seperti ikan. *CNN* memiliki kemampuan untuk mempelajari fitur kompleks secara otomatis dari data mentah, seperti piksel gambar, tanpa perlu ekstraksi fitur manual yang rumit. Keunggulan ini memungkinkan *CNN* untuk memberikan hasil yang sangat baik dalam klasifikasi ikan cupang, dengan akurasi yang cukup tinggi dalam pengenalan jenis ikan. Namun, meskipun *CNN* sangat efektif, arsitektur ini memiliki bias spasial induktif dan cenderung memproses informasi secara lokal. Hal ini membatasi kemampuannya dalam menangkap dependensi jarak jauh yang penting untuk membedakan varietas ikan cupang yang sangat mirip. Meskipun *CNN* sudah menunjukkan hasil yang baik, masih terdapat ruang untuk perbaikan dalam hal akurasi identifikasi, terutama pada varietas ikan cupang yang sangat mirip satu sama lain (Saputra & Agastya, 2024).

Sebagai respons terhadap keterbatasan *CNN*, munculnya *Vision Transformer (ViT)* telah membuka kemungkinan baru dalam klasifikasi citra. *ViT*, yang awalnya dikembangkan untuk pemrosesan bahasa alami (*Natural Language Processing - NLP*), telah berhasil diadaptasi untuk pengolahan citra. *ViT* dapat menangani pola citra yang lebih kompleks dan dependensi jarak jauh secara lebih efektif daripada *CNN*, yang cenderung hanya memproses informasi secara lokal. Penerapan *ViT* dalam klasifikasi ikan cupang sangat relevan, mengingat kemampuannya untuk memahami fitur global yang lebih baik dan menangani kompleksitas *Visual* yang tinggi. Beberapa penelitian sebelumnya menunjukkan bahwa *ViT* telah menunjukkan hasil yang sangat baik dalam mengklasifikasikan jenis ikan lainnya, seperti ikan Koi dengan akurasi 89% dan ikan laut dengan akurasi hingga 97,5%. Meskipun *ViT* memiliki potensi besar dalam klasifikasi citra ikan

cupang, masih terdapat sedikit penelitian yang secara khusus memanfaatkan teknologi *deep learning* berbasis *Vision Transformer* untuk identifikasi ikan cupang secara otomatis. Oleh karena itu, penelitian ini bertujuan untuk mengisi kesenjangan tersebut dan memanfaatkan keunggulan ViT dalam meningkatkan akurasi serta efisiensi identifikasi jenis ikan cupang (Uthama et al., 2024; Yang et al., 2024).

2. METODE PENELITIAN

2.1. Jenis Penelitian

Penelitian ini menggunakan pendekatan penelitian kuantitatif eksperimental. Jenis penelitian ini dipilih karena bertujuan untuk mengukur kinerja dan efektivitas metode *Vision Transformer* (ViT) dalam melakukan klasifikasi citra ikan cupang. Dengan menggunakan data kuantitatif, penelitian ini menguji hipotesis dan mengevaluasi hasil klasifikasi melalui metrik-metrik numerik seperti akurasi, presisi, dan *recall*. Pendekatan yang digunakan adalah *deep learning* dengan memanfaatkan model *pretrained* untuk tugas klasifikasi.

2.2. Akuisisi dan Persiapan *Dataset*

Pada penelitian ini, data yang digunakan untuk melatih dan menguji model klasifikasi ikan cupang terbagi menjadi dua kategori, yaitu data primer dan data sekunder.

a. Data Primer

Data primer diperoleh langsung dari pengumpulan citra ikan cupang yang dilakukan sendiri. Citra-citra ini digunakan untuk menguji dan melatih model klasifikasi guna mengidentifikasi jenis ikan cupang (*Halfmoon*, *Plakat*, dan *Crowntail*). Pengambilan citra ikan cupang dilakukan secara langsung di Pasar Ikan Hias Jatinegara, Jakarta Timur, yang dikenal sebagai salah satu sentra penjualan ikan hias dengan variasi jenis yang cukup lengkap. Data primer ini dipilih karena diambil langsung dari sumber pertama, yakni hasil pengamatan dan dokumentasi yang saya lakukan, sehingga relevansi dan kualitas data lebih terjamin. Fungsi dari pengambilan data gambar tersebut bertujuan untuk menambah akurasi dan juga untuk mengetes fungsi dari identifikasi jenis ikan cupang. Dalam penelitian ini data primer dibagi menjadi dua bagian yaitu yang pertama adalah untuk melengkapi proses *training* dan *validating* data. Untuk bagian ke-dua berfungsi sebagai data uji atau testing terhadap hasil dari training dan validating model yang telah diproses. Berikut adalah jumlah data yang telah dikumpulkan secara mandiri :

Tabel 1. Data Primer

| Jenis Ikan Cupang | Jumlah     |
|-------------------|------------|
| <i>Halfmoon</i>   | 50 gambar  |
| <i>Plakat</i>     | 181 gambar |
| <i>Crowntail</i>  | 83 gambar  |

Data gambar diatas belum melewati proses *Augmentasi*. Dalam proses pengambilan gambar ikan cupang, penulis mengambil gambar dari sisi samping ikan. Hal ini dilakukan agar seluruh bagian tubuh termasuk bentuk sirip, ekor, dan morfologi utama ikan terlihat dengan jelas. Pengambilan dari sisi samping dinilai paling optimal untuk proses klasifikasi karena ciri *visual* tiap jenis ikan cupang seperti *Halfmoon*, *Plakat*, dan *Crowntail* paling mudah dikenali dari bentuk dan orientasi sirip yang tampak dari sisi tersebut.

b. Data Sekunder

Data sekunder diperoleh dari proses pengumpulan gambar ikan cupang yang dilakukan oleh rekan peneliti di Bangladesh, yang kemudian dikombinasikan dengan dataset yang tersedia di Kaggle. (<https://www.kaggle.com/Datasets/myapiti/five-variants-of-beta-splendens-fish>). Proses pengambilan data manual dilakukan dengan mengambil gambar ikan cupang yang telah tersedia di internet,

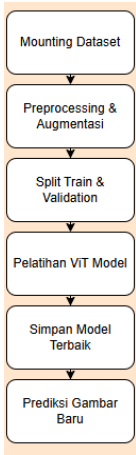
khususnya di *Google Images*, dan digunakan untuk melengkapi Dataset yang ada. Karena data ini diperoleh dari sumber eksternal yang tidak langsung terkait dengan penelitian ini, maka data ini dikategorikan sebagai data sekunder. Penggunaan data sekunder ini bertujuan untuk memperluas variasi gambar ikan cupang dalam Dataset, sehingga dapat meningkatkan kemampuan model dalam melakukan klasifikasi yang lebih akurat. Jumlah data yang diperoleh dari kedua sumber tersebut adalah sebagai berikut :

Tabel 2. Data Sekunder

| Jenis Ikan Cupang | Jumlah     |
|-------------------|------------|
| <i>Halfmoon</i>   | 520 gambar |
| <i>Plakat</i>     | 143 gambar |
| <i>Crowntail</i>  | 226 gambar |

Data diatas belum termasuk proses *Augmentasi*, yang akan dilakukan. Dengan kombinasi antara data primer yang saya kumpulkan secara langsung dan data sekunder yang diperoleh melalui rekan penulis, diharapkan model dapat dilatih dengan *Dataset* yang lebih lengkap dan representatif untuk menghasilkan klasifikasi ikan cupang yang lebih akurat.

2.3. Diagram Alur Penelitian



Gambar 1. Diagram Alur Penelitian

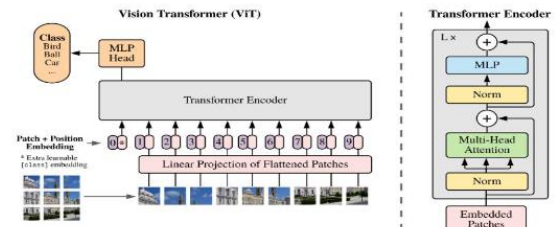
- a. **Mounting Dataset**, Dalam lingkungan komputasi awan seperti *Google Colab*, proses *mounting Dataset* dari penyimpanan eksternal (misalnya *Google Drive*) memungkinkan akses langsung ke data tanpa perlu mengunduhnya ke penyimpanan lokal. Hal ini sangat berguna untuk menangani *Dataset* besar yang tidak muat di memori lokal. Menurut Quaranta et al. (2022), penggunaan *Google Colab* memungkinkan kolaborasi yang efisien dalam pengolahan data, termasuk akses ke *Dataset* melalui *Google Drive*.
- b. **Preprocessing & Augmentasi**, Setelah data diakses, tidak semua data langsung "siap pakai". Tahap ini melibatkan pembersihan dan penyesuaian data. "*Preprocessing*" bisa berarti mengubah ukuran gambar agar seragam, menormalisasi nilai piksel, atau bahkan membersihkan data yang rusak. Sementara itu *Preprocessing* adalah tahap awal dalam pengolahan data yang mencakup normalisasi, perubahan ukuran, dan konversi warna untuk menstandarkan input. Sementara itu, *augmentasi* data bertujuan untuk meningkatkan keragaman data latih dengan menerapkan transformasi seperti rotasi, *flipping*, *zooming*, atau penyesuaian kecerahan guna mencegah *overfitting*. Augmentasi gambar dilakukan untuk memperbesar jumlah data pelatihan dan meningkatkan kemampuan model dalam menggeneralisasi data

yang beragam. Teknik augmentasi yang digunakan meliputi rotasi, *zoom*, *flipping* secara *horizontal*, *rescaling*, serta *random resized crop*. Proses ini bertujuan untuk menghasilkan *Dataset* yang *robust*, yaitu *Dataset* yang lebih bervariasi dan tahan terhadap berbagai perubahan atau gangguan pada citra input, seperti variasi pencahayaan, orientasi, dan ukuran objek (Baso & Suciati, 2020). Dengan melakukan augmentasi gambar, *Dataset* menjadi lebih representatif terhadap kondisi nyata yang mungkin ditemui pada data baru, sehingga model yang dilatih dapat lebih baik dalam menggeneralisasi dan mengenali berbagai variasi citra yang tidak terlihat selama pelatihan. Proses pra-pemrosesan ini juga mencakup normalisasi citra untuk memastikan kestabilan dan konsistensi data yang digunakan dalam pelatihan model. Wang et al. (2024) dalam survei komprehensifnya menyatakan bahwa "*Data augmentation is a series of techniques that generate high-quality Artificial data by manipulating existing data samples.*"

c. **Split Train & Validation**, Ini adalah salah satu tahapan terpenting untuk memastikan model kita belajar dengan baik dan dapat digeneralisasi. *Dataset* yang telah diolah kemudian dibagi menjadi dua bagian: "Data Latih" (untuk melatih model) dan "Data Validasi" (untuk menguji seberapa baik model belajar dari data latih). Membagi *Dataset* menjadi data latih dan validasi adalah praktik umum untuk mengevaluasi performa model dan mencegah *overfitting*. Proporsi umum adalah 80% untuk pelatihan dan 20% untuk validasi. Meskipun ini dapat disesuaikan sesuai kebutuhan, Bai et al. (2020) dalam studi mereka menekankan pentingnya pembagian ini.

d. **Pelatihan Model Vision Transformer**, *Vision Transformer (ViT)* merupakan model *deep learning* yang memproses gambar dengan membaginya menjadi bagian-bagian kecil berukuran tetap, yang kemudian diperlakukan sebagai token untuk dianalisis (Dosovitskiy et al., 2020). Arsitektur *ViT* terdiri dari tiga komponen utama, yaitu *patch + position embedding*, *Transformer encoder*, dan *classification head*. Pada tahap pertama, gambar input dibagi menjadi beberapa *patch* yang memiliki ukuran yang sama, seperti yang terlihat pada Gambar 2. Setiap *patch* diubah menjadi vektor dan kemudian diberikan proyeksi linier untuk menyesuaikan dengan format yang dapat diterima oleh *Transformer encoder*. Setiap vektor *patch* ini juga dilengkapi dengan *positional encoding*, yang memberikan informasi tentang posisi relatif setiap *patch* dalam gambar. Sebuah token kelas tambahan ditambahkan ke dalam urutan untuk bertindak sebagai *placeholder* bagi representasi global gambar yang nantinya akan digunakan dalam klasifikasi (*Extra Learnable [class] Embedding*). Proses berikutnya adalah ekstraksi fitur menggunakan bagian yang disebut *encoder* dari *Transformer*. Encoder ini terdiri dari beberapa blok, di mana setiap blok memiliki dua bagian utama. Bagian pertama adalah *Multi-Head Self-Attention*, yang memungkinkan model untuk melihat dan membandingkan semua *patch* gambar secara simultan. Fitur ini sangat berguna karena membantu model untuk memahami hubungan antar bagian gambar dan menemukan fitur penting yang tersebar di seluruh gambar. Bagian kedua dalam blok ini adalah *MLP (Multi-Output Perceptron)*, yaitu jaringan saraf yang terdiri dari beberapa lapisan dan berfungsi untuk menambah kompleksitas pada proses model, sehingga informasi yang diperoleh dari mekanisme *attention* dapat diproses menjadi representasi yang lebih mendalam. Setiap blok *encoder* juga dilengkapi dengan lapisan normalisasi (*Norm*) yang terletak setelah *self-attention* dan *MLP*, yang membantu menstabilkan pembelajaran model dengan menormalkan distribusi *output*. (Sarasuartha et al., 2024). Setelah semua *patch* gambar diproses oleh *encoder*, informasi yang dihasilkan diproses lebih lanjut oleh *MLP Head*. *MLP Head* ini terdiri dari satu atau lebih lapisan *Neural Network* yang *dense*, menggunakan fungsi *non-linear* untuk mengolah informasi menjadi representasi global gambar, berdasarkan token klasifikasi.

Representasi global ini kemudian digunakan untuk membuat prediksi akhir, yang berupa distribusi probabilitas yang menunjukkan kemungkinan gambar tersebut masuk ke dalam kategori tertentu yang telah ditentukan selama pelatihan. Prediksi ini sangat penting karena menjadi dasar untuk keputusan klasifikasi akhir model, di mana model menentukan kategori yang paling sesuai dengan objek dalam gambar berdasarkan apa yang telah dipelajari sebelumnya (Sarasuartha et al., 2024). Menurut artikel di *viso.ai*, *Vision Transformer* memiliki berbagai aplikasi dalam tugas pengenalan citra seperti deteksi objek, segmentasi, klasifikasi gambar, dan pengenalan tindakan (Boesch, 2023).



Gambar 2. Arsitektur Model *Vision Transformer* (Sarasuartha et al., 2024)

e. **Simpan Model Terbaik**, Setelah model dilatih, kita tidak ingin kehilangan semua kerja keras itu. Tahap ini berarti menyimpan "*snapshot*" dari model yang memiliki kinerja terbaik pada data validasi. Ini penting karena selama pelatihan, kinerja model bisa naik turun. Selama pelatihan, menyimpan bobot model terbaik berdasarkan performa validasi adalah praktik umum untuk menghindari *overfitting*. Hal ini dilakukan melalui mekanisme seperti *checkpointing*. Menurut artikel di Medium oleh Piyush Kashyap, menyimpan dan memuat *checkpoint* selama pelatihan model pembelajaran mendalam adalah aspek penting untuk mempertahankan kemajuan pelatihan (Kashyap, 2024).

f. **Prediksi Gambar Baru**, Ini adalah tujuan akhir dari sistem ini. Setelah model dilatih dan disimpan, kita dapat menggunakannya untuk mengenali atau mengklasifikasikan gambar-gambar yang belum pernah dilihat sebelumnya. Gambar baru akan melalui proses pra-pemrosesan yang serupa dengan data pelatihan, lalu model akan memprosesnya dan memberikan prediksi (misalnya, jenis ikan cupang apa yang ada di gambar tersebut). Ini adalah saatnya kita melihat apakah sistem kita benar-benar "pintar" dan dapat menerapkan pengetahuannya pada kasus-kasus baru.

## 2.4. Perancangan Dataset

*Dataset* merupakan komponen penting dalam sistem klasifikasi ini, terdiri dari gambar-gambar ikan cupang yang dikategorikan berdasarkan spesies atau jenis tertentu. Format *Dataset* mengikuti struktur yang dapat dikenali oleh fungsi *ImageFolder*, yaitu setiap kelas berada dalam *subfolder* yang berbeda, memudahkan proses pemuatan dan pelabelan otomatis oleh *PyTorch*. *Dataset* diunggah ke *Google Drive* dan diakses melalui *Google Colab*. Jumlah kelas dalam *Dataset* ditentukan secara otomatis oleh program berdasarkan jumlah *Folder* di direktori utama *Dataset*.

Untuk mengevaluasi performa model secara komprehensif, *Dataset* ini akan dibagi menjadi data latih dan data validasi dengan beberapa skema rasio yang berbeda. Rasio yang akan diuji meliputi:

- 80% data latih dan 20% data validasi.
- 70% data latih dan 30% data validasi.
- 60% data latih dan 40% data validasi.

Pembagian ini bertujuan untuk mengidentifikasi rasio optimal yang memberikan keseimbangan terbaik antara pembelajaran model yang efektif dan kemampuan generalisasi yang kuat, sekaligus mencegah *overfitting*.

Selanjutnya, dilakukan transformasi citra agar sesuai dengan kebutuhan model *Vision Transformer (ViT)*. Transformasi ini meliputi:

- Untuk *training* set: Dilakukan augmentasi dengan *RandomResizedCrop* dan *RandomHorizontalFlip* untuk meningkatkan variasi data. Citra kemudian dikonversi ke tensor dan dinormalisasi berdasarkan nilai mean dan standar deviasi *ImageNet*.
- Untuk *validation* set: Dilakukan *resizing* dan *cropping* ke ukuran standar, kemudian dikonversi ke *tensor* dan dinormalisasi.

Transformasi ini membantu menyesuaikan dimensi dan distribusi data agar sesuai dengan model *pre-trained Vision Transformer* yang digunakan.

#### 2.4.1. Pengunggahan Dataset Citra ke Google Drive

*Dataset* citra ikan cupang yang telah dikumpulkan dari berbagai sumber, termasuk koleksi pribadi dan hasil pencarian daring, diunggah ke dalam *Google Drive*. Langkah ini dilakukan untuk memastikan bahwa *Dataset* dapat diakses secara efisien dan terintegrasi dengan platform lain, seperti *Google Colab*, yang digunakan dalam pelatihan model. Mengapa *Google Drive* dipilih? Penggunaan *Google Drive* memungkinkan penyimpanan data dalam jumlah besar secara aman dan fleksibel, serta mempermudah proses pengelolaan dan pemrosesan data dari berbagai lokasi. Selain itu, *Google Drive* juga menyediakan integrasi langsung dengan *Google Colab*, sehingga memudahkan proses *loading* dan penggunaan data dalam lingkungan komputasi berbasis *cloud*. Penyimpanan di *Google Drive* juga memastikan bahwa data dapat diakses kapan saja dan di mana saja, mendukung kelancaran pengembangan model dalam skala besar.

#### 2.4.2. Pembagian Dataset menjadi Data Latih dan Data Validasi

*Dataset* yang telah dimuat selanjutnya akan dibagi ke dalam dua bagian utama: data pelatihan dan data validasi. *Dataset* citra ikan cupang dibagi menjadi data pelatihan dan data validasi dengan rasio seperti 80:20, 70:30, dan 60:40 untuk menguji ketahanan dan performa model secara komprehensif. Pendekatan ini penting untuk mencegah *overfitting* dan memastikan evaluasi performa model yang objektif (Murphy, 2022). Proses pembagian dilakukan secara acak dengan menjaga distribusi kelas yang seimbang di setiap *subset* data, sesuai dengan praktik standar dalam *machine learning*.

#### 2.4.3. Transformasi dan Augmentasi Data

Data pelatihan diolah dengan teknik augmentasi untuk meningkatkan keberagaman data, sehingga model dapat mempelajari fitur yang lebih robust dan general. Teknik augmentasi yang digunakan meliputi *RandomResizedCrop* dan *RandomHorizontalFlip* (Cubuk et al., 2020). *RandomResizedCrop* memotong dan mengubah ukuran gambar secara acak, memungkinkan model mengenali ikan cupang dalam berbagai skala dan posisi. *RandomHorizontalFlip* membalik gambar secara horizontal secara acak, membantu model mengenali ikan cupang dari sudut pandang berbeda, sehingga meningkatkan kemampuan generalisasi model terhadap variasi orientasi gambar. *RandomResizedCrop(x)* dan *RandomHorizontalFlip(x)* adalah dua teknik augmentasi utama yang digunakan dalam sistem ini. Rumus matematikanya adalah sebagai berikut:

$$x' = \text{RandomResizedCrop}(x) \text{ dan } x'' = \text{RandomHorizontalFlip}(x')$$

Di mana:

- $x$  adalah gambar input,
- $x'$  adalah gambar setelah dipotong dan diubah ukurannya secara acak, dan
- $x''$  adalah gambar yang dibalik secara horizontal. (Touvron et al., 2020)

Untuk data validasi, prosesnya lebih sederhana. Hanya *resizing* dan normalisasi citra yang dilakukan. *Resizing* memastikan semua gambar validasi memiliki ukuran yang seragam, sama seperti data pelatihan. Selain itu, gambar juga dinormalisasi berdasarkan mean dan standar deviasi *Dataset ImageNet*, menggunakan rumus berikut:

$$x_{\text{normalized}} = \frac{x - \mu}{\sigma}$$

Di mana:

- $\mu = [0.485, 0.456, 0.406]$  adalah nilai rata-rata RGB *ImageNet*,
- $\sigma = [0.229, 0.224, 0.225]$  adalah standar deviasi RGB *ImageNet* (Goodfellow et al., n.d.)

Normalisasi, di sisi lain, menyesuaikan rentang nilai piksel agar model dapat memproses gambar dengan lebih stabil dan konsisten. Proses sederhana pada data validasi ini penting untuk menjaga konsistensi evaluasi, memastikan hasil pengujian model tetap objektif dan tidak terpengaruh oleh variasi augmentasi yang diterapkan pada data pelatihan. Pra-pemrosesan data seperti ini adalah langkah vital dalam *machine learning* untuk mengubah data mentah menjadi format yang bersih dan terstruktur untuk pelatihan model, meningkatkan akurasi, dan mencegah data leakage.

#### 2.5. Perancangan Model

Model yang digunakan dalam sistem ini adalah *Vision Transformer (ViT)*, tepatnya varian *vit\_base\_patch16\_224* yang merupakan salah satu model stabil dan populer dalam library *timm*. *Vision Transformer*, yang awalnya dirancang untuk pemrosesan bahasa alami, telah dimodifikasi untuk digunakan dalam tugas pengolahan citra, dan terbukti efektif dalam mengenali pola *Visual* pada gambar.

Langkah-langkah dalam perancangan model adalah sebagai berikut:

- Memuat Model *ViT* yaitu, Pertama, model *ViT* yang telah dilatih sebelumnya pada *Dataset ImageNet* dimuat ke dalam sistem. Penggunaan model *pre-trained* ini memungkinkan pemanfaatan pengetahuan umum yang telah diperoleh dari *Dataset* besar, yang dapat mempercepat dan meningkatkan akurasi pelatihan pada *Dataset* yang lebih kecil, seperti *Dataset* ikan cupang.
- Mengganti *Output* Klasifikasi, *Output* terakhir atau *classification head* dari model *pre-trained* ini diganti agar sesuai dengan jumlah kelas pada *Dataset* ikan cupang. Secara default, model *ViT* dilatih untuk 1000 kelas pada *Dataset ImageNet*, sehingga *output* klasifikasi akhir tersebut diubah menjadi *nn.Linear(num\_ftrs, num\_classes)* untuk menyesuaikan dengan tiga kelas yang ada dalam *Dataset (Halfmoon, Plakat, Crowntail)*. Berikut adalah rumus yang digunakan untuk memodifikasi *output* klasifikasi (*head classifier*) dari model *ViT*:

$$\text{Output} = \text{Softmax}(W \cdot x + b)$$

Di mana:

- $W$  adalah bobot pada *output* terakhir,
  - $x$  adalah representasi citra yang diproses oleh model,
  - $b$  adalah bias pada *output* terakhir.
- Softmax* mengubah *output* model menjadi probabilitas kelas.
- Optimasi dengan *Adam* adalah proses model ini dioptimalkan menggunakan *Adam Optimizer*. *Adam* merupakan algoritme optimisasi yang efektif untuk pelatihan model *deep learning*, dengan kemampuan untuk menyesuaikan *learning rate* secara adaptif untuk setiap parameter. *Adam* menggabungkan kelebihan dari Momentum dan *RMSProp*, yang memungkinkan model untuk konvergen lebih cepat dan menghindari masalah *vanishing gradient*. *Optimizer* ini menghitung estimasi pertama (*mean*) dan kedua (*uncentered variance*) dari gradien, dan menggunakan informasi ini untuk mengatur pembaruan bobot dalam setiap iterasi. Dengan pengaturan *learning rate* yang optimal dan penggunaan estimasi gradien yang adaptif, *Adam* dapat mempercepat proses pelatihan model *Vision Transformer* ini (Wulaning Ayu & Pradipta, 2022).
  - Pindah ke *GPU (Graphic Processing Unit)*, Setelah *output* klasifikasi dimodifikasi dan *optimizer* ditetapkan, model kemudian dipindahkan ke perangkat *GPU* menggunakan perintah *model.to(device)*. Pemanfaatan *GPU* memungkinkan proses pelatihan dilakukan lebih cepat, terutama pada *Dataset* besar yang

memerlukan komputasi intensif, seperti yang digunakan dalam model *ViT*.

Dengan memanfaatkan model *pre-trained ViT* dan optimasi menggunakan *Adam*, sistem ini dapat memanfaatkan pengetahuan yang sudah dipelajari dari *Dataset ImageNet*, sehingga pelatihan pada *Dataset* ikan cupang menjadi lebih efisien dan akurat.

## 2.6. Perancangan Proses Pelatihan

Proses pelatihan merupakan inti dari sistem klasifikasi ini. Tujuan utama dari pelatihan adalah untuk mengoptimalkan bobot dalam jaringan *Vision Transformer (ViT)* sehingga model dapat memetakan input citra ke kelas yang benar dengan tingkat akurasi yang tinggi. Proses pelatihan dilakukan selama beberapa *epoch*, dengan setiap *epoch* terdiri dari dua fase utama, yaitu pelatihan dan validasi.

Langkah-langkah dalam proses pelatihan adalah sebagai berikut:

### 1. Inisialisasi Komponen

- Fungsi *Loss* yaitu fungsi yang digunakan untuk menghitung perbedaan antara *output model* dan label target adalah *Cross-EntropyLoss*. Fungsi ini umum digunakan dalam klasifikasi multi-kelas dan mengukur kesalahan prediksi model berdasarkan distribusi probabilitas yang dihasilkan. Model dilatih menggunakan *Cross-EntropyLoss* untuk klasifikasi multi-kelas, dengan rumus sebagai berikut:

$$L = - \sum_{i=1}^C y_i \log(p_i)$$

Di mana:

- $L$  adalah nilai loss,
- $C$  adalah jumlah kelas (3 jenis ikan cupang),
- $y_i$  adalah label target untuk kelas ke- $i$ ,
- $p_i$  adalah probabilitas prediksi untuk kelas ke- $i$
- Optimizer* yang digunakan, *Adam* adalah algoritma optimisasi adaptif yang menggabungkan kelebihan dari Momentum dan *RMSProp*, yang memungkinkan model untuk mencapai konvergensi lebih cepat. *Adam* menyesuaikan *learning rate* untuk setiap parameter secara dinamis, yang memudahkan model untuk beradaptasi dengan perubahan gradien pada setiap iterasi. *Learning rate* yang digunakan adalah  $1e-4$ , yang merupakan pengaturan umum yang sering digunakan untuk model *ViT*. Proses pelatihan ini akan dilakukan selama 15 *epoch* dengan harapan model dapat mencapai konvergensi yang optimal. Berikut adalah rumus yang digunakan oleh *Adam Optimizer* (Sarasuartha et al., 2024) :

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) \cdot g_t \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) \cdot g_t^2 \\ \hat{m}_t &= \frac{m_t}{1 - \beta_1^t} \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \\ \theta_{t+1} &= \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \end{aligned}$$

Di mana:

- $m_t$  adalah estimasi momen pertama (mean) dari gradien,
- $v_t$  adalah estimasi momen kedua (variansi tak terpusat) dari gradien,
- $g_t$  adalah gradien pada waktu  $t$
- $\beta_1$  dan  $\beta_2$  adalah koefisien peluruhan eksponensial untuk estimasi momen,
- $\theta$  adalah parameter yang akan dioptimalkan,
- $\alpha$  adalah *learning rate*, dan
- $\epsilon$  adalah konstanta kecil untuk mencegah pembagian dengan nol

### 2. Scheduler

*StepLR* digunakan untuk menurunkan *learning rate* secara bertahap setelah sejumlah *epoch* tertentu. Penurunan *learning rate* ini membantu model untuk menyesuaikan pembelajaran pada fase akhir pelatihan,

mencegah *overfitting* dan membantu model mencapai hasil yang lebih optimal.

### 3. Training Phase

- Model diatur dalam mode pelatihan menggunakan *model.train()*.
- Setiap batch data yang diambil dari *train\_loader* digunakan untuk melakukan *forward pass*, di mana gambar input akan diproses oleh model untuk menghasilkan prediksi.
- Selanjutnya, *loss* dihitung berdasarkan prediksi model dan label target. *Backward propagation* dilakukan untuk menghitung gradien yang diperlukan untuk memperbarui bobot model.
- Optimizer Adam* digunakan untuk memperbarui bobot model berdasarkan gradien yang dihitung. Pembaruan bobot dilakukan secara adaptif sesuai dengan estimasi gradien yang dihitung oleh *Adam*.

### 4. Validation Phase

- Setelah setiap *epoch* pelatihan, model akan diuji pada data validasi dengan mengatur model dalam mode evaluasi menggunakan *model.eval()*.
- Data validasi digunakan untuk mengukur kinerja model tanpa melakukan pembaruan bobot, sehingga dapat memberikan gambaran yang objektif mengenai bagaimana model bekerja dengan data yang belum pernah dilihat selama pelatihan.
- Pada fase ini, nilai *loss* dan akurasi dihitung untuk mengevaluasi seberapa baik model dapat menggeneralisasi pada data yang tidak terlibat dalam proses pelatihan.

### 5. Penyimpanan Model Terbaik, adalah :

- Setiap kali akurasi validasi pada suatu *epoch* lebih tinggi dari *epoch* sebelumnya, model akan disimpan menggunakan mekanisme *checkpointing*. Model terbaik, yang memiliki kinerja tertinggi pada data validasi, akan disimpan dalam format *.pth* untuk digunakan pada tahap inferensi.
- Penyimpanan model yang optimal sangat penting untuk memastikan bahwa model yang sudah terlatih dengan baik dapat digunakan kembali tanpa perlu melatih ulang dari awal.
- Selama pelatihan, nilai akurasi dan *loss* dari setiap *epoch* akan disimpan dalam *dictionary history*. Nilai-nilai ini nantinya dapat digunakan untuk *Visualisasi* hasil implementasi dan pengujian sistem.

### 6. Penggunaan Model untuk Melakukan Prediksi terhadap Citra Baru

- Setelah pelatihan selesai, kita dapat menguji model dengan citra baru yang belum ada dalam *Dataset* pelatihan.
- Gambar baru ini akan melalui beberapa tahap pra-pemrosesan terlebih dahulu, seperti *resizing*, normalisasi, dan transformasi yang diperlukan agar gambar sesuai dengan format yang dibutuhkan oleh model.
- Setelah proses pra-pemrosesan, citra tersebut akan diproses oleh model yang telah dilatih untuk memprediksi kelasnya.
- Model kemudian akan memberikan hasil berupa nama jenis ikan cupang yang terdeteksi, beserta dengan probabilitasnya, yang menunjukkan sejauh mana model yakin dengan prediksi tersebut. Fungsi ini sangat berguna untuk pengujian manual atau verifikasi, terutama untuk memastikan bahwa model dapat memberikan hasil yang akurat pada citra yang belum dilatih sebelumnya.

Selama pelatihan, nilai akurasi dan *loss* dari setiap *epoch* akan disimpan dalam *dictionary history*. Nilai-nilai ini nantinya dapat digunakan untuk *Visualisasi* hasil implementasi dan pengujian sistem. Dengan memanfaatkan *optimizer Adam*, yang memiliki kemampuan untuk menyesuaikan *learning rate* secara adaptif dan meningkatkan efisiensi pelatihan, diharapkan model dapat mencapai konvergensi lebih cepat dan menghasilkan prediksi yang lebih akurat.

## 2.7. Perancangan Fungsi Prediksi

Setelah model dilatih dan disimpan, sistem menyediakan fungsi inferensi untuk melakukan klasifikasi terhadap gambar baru. Fungsi prediksi ini bekerja dengan langkah-langkah berikut:

- Gambar dibuka menggunakan library *PIL* dan dikonversi menjadi *RGB*.
- Transformasi yang sama dengan data validasi diaplikasikan untuk memastikan konsistensi.
- Gambar diubah menjadi *tensor* dan diberi dimensi *batch* (*unsqueeze(0)*).
- Gambar dikirim ke perangkat *GPU* (*Graphic Processing Unit*) jika tersedia.
- Model melakukan *forward pass* terhadap input gambar.
- Hasil prediksi berupa probabilitas dikonversi menjadi label kelas dengan nilai tertinggi.

Hasil prediksi ditampilkan bersama dengan gambar input dan informasi probabilitas. Fungsi ini memungkinkan pengguna untuk menguji performa model pada citra baru secara *real-time* tanpa harus melatih ulang model.

## 2.8. Penangan *Error* dan Validasi *Dataset*

Selama proses pengembangan sistem identifikasi jenis ikan cupang berbasis *Vision Transformer*, penting untuk memastikan bahwa sistem mampu menangani berbagai kemungkinan kesalahan (*error*) yang dapat terjadi, baik dari sisi teknis maupun kualitas data. Oleh karena itu, dilakukan penerapan berbagai metode penanganan *error* serta validasi *dataset* untuk meningkatkan keandalan dan ketahanan sistem terhadap kondisi yang tidak ideal. Langkah ini tidak hanya bertujuan untuk mencegah kegagalan sistem secara tiba-tiba, tetapi juga untuk memberikan petunjuk atau pesan kesalahan yang jelas kepada pengguna agar dapat segera mengambil tindakan korektif.

### 2.8.1. Validasi *Path Dataset*

Salah satu *error* umum yang diantisipasi dalam sistem ini adalah ketika *path* atau jalur direktori *dataset* tidak ditemukan. Sebelum proses pemuatan data dimulai, sistem akan terlebih dahulu memeriksa apakah jalur yang diberikan benar-benar ada dan dapat diakses. Jika *path* tidak valid misalnya direktori tidak ada, salah ketik, atau *file* telah dipindahkan sistem akan secara otomatis menghentikan proses dan menampilkan pesan *error* yang eksplisit, seperti: “*Error: Direktori dataset tidak ditemukan. Pastikan path sudah benar.*” Penanganan ini bertujuan untuk mencegah *crash* mendadak dan memberikan umpan balik langsung kepada pengguna agar dapat segera memperbaiki kesalahan. Dengan adanya validasi *path* ini, sistem menjadi lebih stabil dan tahan terhadap kesalahan input dari sisi pengguna.

### 2.8.2. Validasi Struktur dan Format *Dataset*

Masalah lain yang sering muncul berkaitan dengan struktur *folder dataset* yang tidak sesuai. Sistem ini menggunakan pustaka *PyTorch* dengan class *ImageFolder*, yang mengharuskan dataset memiliki struktur direktori tertentu. Dalam format ini, *folder* utama dataset harus berisi *subfolder* yang masing-masing mewakili satu kelas atau label (misalnya *Halfmoon*, *Crowntail*, dan *Plakat*), dan setiap *subfolder* tersebut harus berisi file gambar dengan format yang didukung seperti *.jpg* atau *.png*. Jika struktur ini tidak terpenuhi misalnya seluruh gambar dicampur dalam satu *folder*, tidak ada *subfolder*, atau *subfolder* kosong maka sistem tidak dapat memuat data secara benar.

Untuk mengatasi potensi *error* ini, sistem dilengkapi dengan mekanisme pengecekan struktur direktori sebelum proses pelatihan dimulai. Jika ditemukan ketidaksesuaian, sistem akan menampilkan pesan peringatan seperti: “*Peringatan: Struktur dataset tidak sesuai format ImageFolder. Pastikan setiap kelas memiliki subdirektori yang berisi gambar.*” Selain itu, sistem juga dirancang untuk mengabaikan file gambar yang rusak atau tidak dapat dibuka—seperti gambar korup atau berekstensi tidak sesuai. Jika ditemukan file yang bermasalah, sistem tidak akan menghentikan seluruh proses, melainkan melewatinya

dan mencatat pesan peringatan dalam *log*, seperti: “*Peringatan: Gambar [nama\_file.jpg] tidak dapat dibuka dan akan dilewati.*” Dengan validasi struktur dan format seperti ini, sistem menjadi lebih tangguh dalam menangani anomali data dan tetap mampu melanjutkan proses pelatihan meskipun ada sebagian file yang bermasalah.

## 3. HASIL DAN PEMBAHASAN

### 3.1. Analisis Hasil Uji dengan Berbagai Rasio dan Pengujian

Setelah dilakukan proses pengujian terhadap model *Vision Transformer* menggunakan beberapa gambar ikan cupang, langkah selanjutnya adalah melakukan analisis terhadap hasil yang diperoleh. Analisis ini bertujuan untuk mengevaluasi performa model dalam mengklasifikasikan gambar-gambar baru secara *Visual* maupun numerik berdasarkan probabilitas yang dihasilkan.

Untuk menguji konsistensi performa model *Vision Transformer* (*ViT*) dalam mengidentifikasi jenis ikan cupang, dilakukan eksperimen dengan menggunakan empat rasio pembagian data latih dan data uji yang berbeda, yaitu 60:40, 70:30, 80:20, dan 50:50. Tujuan dari eksperimen ini adalah untuk mengetahui sejauh mana perubahan proporsi data dapat memengaruhi akurasi, *loss*, dan metrik evaluasi lainnya seperti *precision*, *recall*, dan *F1-score*.

Pada setiap rasio, data dibagi menggunakan fungsi *train\_test\_split* atau pendekatan yang setara pada *torchVision.Datasets.ImageFolder* dengan pengacakan (*shuffle=True*) agar proses pemisahan data tetap adil dan acak. Model kemudian dilatih ulang dari awal pada masing-masing konfigurasi rasio, menggunakan parameter pelatihan yang sama untuk menjaga keadilan dalam perbandingan.

Selama pelatihan pada masing-masing rasio, dicatat nilai akurasi dan *loss* baik pada data latih maupun validasi. Setelah model mencapai performa terbaik, model disimpan dan kemudian diuji menggunakan data uji yang sesuai dengan masing-masing rasio. Evaluasi dilakukan menggunakan *confusion matrix* dan perhitungan metrik evaluasi lainnya.

Secara umum, semakin besar rasio data latih, model memiliki peluang lebih besar untuk belajar pola yang kompleks, namun dapat mengurangi jumlah data uji yang tersedia untuk evaluasi. Sebaliknya, rasio data uji yang besar memungkinkan evaluasi yang lebih kuat namun dapat menurunkan performa pelatihan karena keterbatasan data latih.

#### 1. Pengujian dengan Rasio 80:20

Pada pengujian ini, *Dataset* dibagi dengan rasio 80% untuk pelatihan dan 20% untuk pengujian. Total sebanyak 1.203 citra ikan cupang digunakan, yang terdiri dari tiga kelas, yaitu *Crowntail*, *Halfmoon*, dan *Plakat*. Dengan pembagian ini, sebanyak 962 citra digunakan untuk pelatihan dan 241 citra digunakan untuk pengujian.

Model *Vision Transformer* (*ViT*) yang digunakan adalah *vit\_base\_patch16\_224*, dan telah dimodifikasi untuk mengklasifikasikan tiga kelas. Proses pelatihan dilakukan selama 15 *epoch* menggunakan *GPU* dengan dukungan *CUDA* (*Compute Unified Device Architecture*), yaitu sebuah platform komputasi paralel dari *NVIDIA* yang memungkinkan percepatan proses perhitungan model deep learning secara signifikan dibandingkan dengan penggunaan *CPU* biasa. Dengan memanfaatkan *CUDA*, proses pelatihan dapat berjalan lebih cepat dan efisien karena mampu menjalankan banyak operasi matriks secara paralel pada ribbon inti *GPU*. Model terbaik disimpan secara otomatis ketika akurasi validasi tertinggi tercapai, yaitu pada *epoch* ke-6 dengan akurasi validasi sebesar 98,34% dan nilai *loss* sebesar 0.1100.

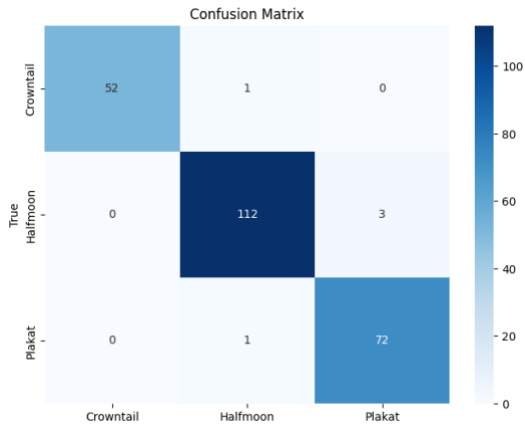
Selama proses pelatihan, performa model meningkat signifikan. Akurasi pelatihan mencapai 99,79% pada *epoch* ke-6. Setelah pelatihan selesai, model terbaik digunakan untuk menguji performa terhadap data uji. Evaluasi dilakukan menggunakan metrik

precision, recall, f1-score, dan support, seperti ditampilkan pada Tabel 3 berikut:

Tabel 3. *Classification Report* Pengujian Rasio 80:20

| Kelas            | Precision | Recall | F1-Score | Support |
|------------------|-----------|--------|----------|---------|
| <i>Crowntail</i> | 1.00      | 0.98   | 0.99     | 53      |
| <i>Halfmoon</i>  | 0.98      | 0.97   | 0.98     | 115     |
| <i>Plakat</i>    | 0.96      | 0.99   | 0.97     | 73      |
| accuracy         | -         | -      | 0.98     | 241     |
| macro avg        | 0.98      | 0.98   | 0.98     | 241     |
| weighted avg     | 0.98      | 0.98   | 0.98     | 241     |

Selanjutnya, *Confusion Matrix* ditampilkan untuk menunjukkan persebaran prediksi terhadap masing-masing kelas secara visual pada Gambar 3.



Gambar 3. *Confusion Matrix* Pengujian Rasio 80:20

Berdasarkan hasil evaluasi, model *Vision Transformer* menunjukkan performa klasifikasi yang sangat baik. Kelas *Crowntail* diklasifikasikan hampir sempurna, dengan precision sebesar 1.00 dan recall 0.98. Kelas *Halfmoon* dan *Plakat* juga menunjukkan performa yang sangat baik, meskipun terdapat sedikit kesalahan klasifikasi. Akurasi keseluruhan mencapai 98%, yang mengindikasikan bahwa model mampu mengenali karakteristik visual dari masing-masing jenis ikan cupang secara efektif. Performa tinggi ini kemungkinan besar disebabkan oleh proporsi data latih yang besar (80%), sehingga model memiliki cukup informasi untuk membentuk representasi fitur yang baik dan melakukan generalisasi terhadap data uji.

Berdasarkan *confusion matrix* yang ditunjukkan pada Gambar 4.18, dapat dilihat bahwa sistem mampu mengklasifikasikan gambar ikan cupang ke dalam tiga kelas utama yaitu *Crowntail*, *Halfmoon*, dan *Plakat* dengan akurasi yang tinggi. Hal ini ditunjukkan oleh dominasi nilai pada diagonal utama matriks, yaitu: 52 gambar *Crowntail* terklasifikasi benar sebagai *Crowntail*, 112 gambar *Halfmoon* terklasifikasi benar sebagai *Halfmoon*, dan 72 gambar *Plakat* terklasifikasi benar sebagai *Plakat*.

Namun demikian, masih terdapat beberapa kesalahan klasifikasi (*misclassification*) yang muncul, di antaranya: sebanyak 1 gambar *Crowntail* diklasifikasikan sebagai *Halfmoon*, kemungkinan karena kemiripan karakteristik visual antara ikan *Crowntail* dan *Halfmoon*, seperti bentuk ekor yang melengkung atau variasi warna yang serupa. Sebanyak 1 gambar *Plakat* juga diklasifikasikan sebagai *Halfmoon*, yang dapat disebabkan oleh sudut pengambilan gambar atau kualitas gambar yang kurang optimal. Selain itu, terdapat 3 gambar *Halfmoon* yang diklasifikasikan sebagai *Plakat*, yang kemungkinan terjadi akibat kesamaan bentuk tubuh atau posisi sirip yang tidak tertangkap secara jelas oleh model.

Kesalahan klasifikasi tersebut masih dalam batas yang wajar, mengingat model bekerja berdasarkan informasi visual yang terkadang sulit dibedakan bahkan oleh manusia. Secara keseluruhan, jumlah kesalahan relatif kecil dibandingkan jumlah prediksi yang benar,

sehingga performa model *Vision Transformer* dalam mengidentifikasi jenis ikan cupang dapat dinilai sangat baik.

## 2. Pengujian dengan Rasio 70:30

Pada pengujian kedua ini, *Dataset* dibagi dengan rasio 70% untuk pelatihan dan 30% untuk pengujian, dari total 1.203 citra ikan cupang. Pembagian ini menghasilkan 842 citra untuk pelatihan dan 361 citra untuk pengujian. Sama seperti sebelumnya, data terdiri dari tiga kelas, yaitu *Crowntail*, *Halfmoon*, dan *Plakat*.

Model yang digunakan tetap menggunakan arsitektur *vit\_base\_patch16\_224*, yang telah disesuaikan untuk klasifikasi tiga kelas. Proses pelatihan dilakukan selama 15 *epoch* menggunakan *GPU (CUDA)*. Model terbaik disimpan secara otomatis saat mencapai akurasi validasi tertinggi, yaitu pada *epoch* ke-6 dengan akurasi validasi sebesar 98,34% dan nilai loss sebesar 0.0486.

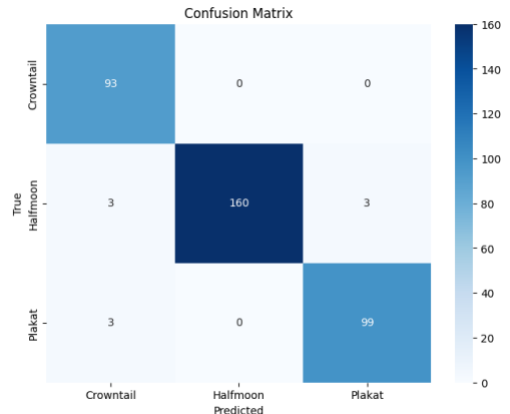
Selama proses pelatihan, model menunjukkan peningkatan performa yang cepat. Akurasi pelatihan meningkat hingga mencapai 99,88% mulai dari *epoch* ke-8 dan seterusnya, menunjukkan bahwa model berhasil mengenali pola dari data latih secara konsisten.

Evaluasi model terhadap data uji dilakukan setelah proses pelatihan selesai, dengan hasil yang dirangkum dalam Tabel 4.2 berikut:

Tabel 4. *Classification Report* Pengujian Rasio 70:30

| Kelas            | Precision | Recall | F1-Score | Support |
|------------------|-----------|--------|----------|---------|
| <i>Crowntail</i> | 0.94      | 1.00   | 0.97     | 93      |
| <i>Halfmoon</i>  | 1.00      | 0.96   | 0.98     | 166     |
| <i>Plakat</i>    | 0.97      | 0.97   | 0.97     | 102     |
| Accuracy         | -         | -      | 0.98     | 361     |
| Macro Avg        | 0.97      | 0.98   | 0.97     | 361     |
| Weighted Avg     | 0.98      | 0.98   | 0.98     | 361     |

Visualisasi performa model juga ditampilkan dalam bentuk *Confusion Matrix* pada Gambar 4.



Gambar 4. *Confusion Matrix* Pengujian Rasio 70:30

Berdasarkan hasil evaluasi, model menunjukkan performa klasifikasi yang sangat tinggi. Akurasi keseluruhan mencapai 98%, dengan nilai precision dan recall yang tinggi di semua kelas. Kelas *Crowntail* diklasifikasikan dengan sempurna (recall 1.00), sedangkan kelas *Halfmoon* dan *Plakat* hanya mengalami sedikit kesalahan klasifikasi. Kinerja yang optimal ini menunjukkan bahwa model *Vision Transformer* tetap mampu melakukan generalisasi dengan sangat baik meskipun jumlah data pelatihan dikurangi menjadi 70%. Hal ini memperlihatkan ketahanan model terhadap variasi distribusi data selama proses pelatihan dan validasi.

Berdasarkan *confusion matrix* yang ditunjukkan pada Gambar 4.19, sistem mampu mengklasifikasikan gambar ikan cupang ke dalam tiga kelas utama, yaitu *Crowntail*, *Halfmoon*, dan *Plakat* dengan akurasi yang sangat tinggi. Hasil klasifikasi menunjukkan bahwa 93 gambar *Crowntail* terklasifikasi benar sebagai *Crowntail*, 160 gambar *Halfmoon* terklasifikasi benar sebagai *Halfmoon*, dan 99 gambar *Plakat* terklasifikasi benar sebagai *Plakat*.

Meskipun sebagian besar klasifikasi dilakukan dengan benar, masih terdapat beberapa kesalahan klasifikasi yang bersifat minor, di antaranya: 3 gambar *Halfmoon* diklasifikasikan sebagai *Crowntail*, dan 3 gambar lainnya diklasifikasikan sebagai *Plakat*. Hal ini dapat terjadi karena adanya kemiripan visual antara kelas *Halfmoon* dengan dua kelas tersebut, misalnya dari segi bentuk sirip yang tumpang tindih, pencahayaan yang tidak merata, atau sudut pengambilan gambar. Selain itu, 3 gambar *Plakat* juga terklasifikasi sebagai *Crowntail*, yang kemungkinan disebabkan oleh bentuk sirip atau postur tubuh yang menyerupai kelas tersebut pada gambar tertentu.

Kesalahan-kesalahan ini wajar terjadi dalam pemrosesan citra karena model bekerja berdasarkan fitur visual yang terkadang sulit dibedakan secara konsisten. Namun, jumlah kesalahan tersebut relatif sangat kecil dibandingkan dengan total jumlah prediksi, sehingga dapat disimpulkan bahwa model *Vision Transformer* memberikan hasil klasifikasi yang sangat andal dan efektif pada data uji dengan rasio pelatihan 70%.

### 3. Pengujian dengan Rasio 60:40

Pada pengujian ketiga, *Dataset* dibagi dengan rasio 60% untuk pelatihan dan 40% untuk pengujian, dari total 1.203 citra ikan cupang. Pembagian ini menghasilkan 721 citra untuk pelatihan dan 482 citra untuk pengujian. Model yang digunakan masih menggunakan arsitektur *vit\_base\_patch16\_224*, yang dimodifikasi untuk mengenali tiga kelas: *Crowntail*, *Halfmoon*, dan *Plakat*.

Pelatihan dilakukan selama 15 *epoch* menggunakan *GPU* (*CUDA*). Model menunjukkan peningkatan performa secara cepat, dan model terbaik tersimpan secara otomatis pada *epoch* ke-13, dengan akurasi validasi sebesar 96,89% dan loss sebesar 0.0917.

Selama proses pelatihan, akurasi pelatihan meningkat hingga 100% pada *epoch* terakhir, menunjukkan bahwa model berhasil belajar dengan sangat baik dari data pelatihan. Meskipun data latih lebih sedikit dibandingkan pengujian sebelumnya, performa validasi masih tinggi dan stabil.

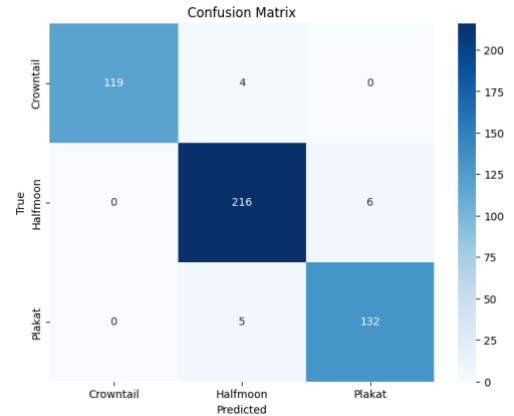
Evaluasi terhadap data uji menghasilkan metrik klasifikasi seperti yang ditampilkan dalam Tabel 4.

Tabel 4. *Classification Report* Pengujian Rasio 60:40

| Kelas               | Precision | Recall | F1-Score | Support |
|---------------------|-----------|--------|----------|---------|
| <i>Crowntail</i>    | 1.00      | 0.97   | 0.98     | 123     |
| <i>Halfmoon</i>     | 0.96      | 0.97   | 0.97     | 222     |
| <i>Plakat</i>       | 0.96      | 0.96   | 0.96     | 137     |
| <i>Accuracy</i>     | –         | –      | 0.97     | 482     |
| <i>Macro Avg</i>    | 0.97      | 0.97   | 0.97     | 482     |
| <i>Weighted Avg</i> | 0.97      | 0.97   | 0.97     | 482     |

Berdasarkan tabel di atas, model menunjukkan performa yang konsisten dalam melakukan klasifikasi pada tiga kelas ikan cupang. Meskipun nilai precision untuk kelas *Plakat* sedikit lebih rendah dibanding dua kelas lainnya, nilai *recall*-nya tinggi, yang menunjukkan bahwa model cukup baik dalam mengenali gambar *Plakat* meskipun mungkin ada beberapa prediksi yang salah diarahkan.

Untuk memperjelas distribusi prediksi yang benar dan salah antar kelas, berikut ditampilkan *confusion matrix* dari pengujian dengan rasio 60:40:



Gambar 5. *Confusion Matrix* Pengujian Rasio 60:40

Berdasarkan hasil evaluasi, model menunjukkan performa klasifikasi yang sangat tinggi. Akurasi keseluruhan mencapai 98%, dengan nilai precision dan recall yang tinggi di semua kelas. Kelas *Halfmoon* diklasifikasikan hampir sempurna (*recall* mendekati 1.00), sedangkan kelas *Crowntail* dan *Plakat* hanya mengalami sedikit kesalahan klasifikasi. Kinerja yang optimal ini menunjukkan bahwa model *Vision Transformer* tetap mampu melakukan generalisasi dengan sangat baik meskipun jumlah data pelatihan dikurangi menjadi 60%. Hal ini memperlihatkan bahwa model tetap tangguh dalam mengenali pola visual meskipun dengan data pelatihan yang lebih terbatas.

Berdasarkan *confusion matrix* pada Gambar 5, sistem mengklasifikasikan gambar ikan cupang ke dalam tiga kelas utama *Crowntail*, *Halfmoon*, dan *Plakat* dengan akurasi tinggi. Sebanyak 119 gambar *Crowntail* berhasil diklasifikasikan dengan benar, begitu juga dengan 216 gambar *Halfmoon* dan 132 gambar *Plakat* yang juga diklasifikasikan sesuai dengan kelas aslinya. Nilai-nilai tersebut mendominasi diagonal utama matriks, menandakan bahwa sebagian besar prediksi yang dilakukan oleh model adalah benar.

Kesalahan klasifikasi masih tetap ada, namun dalam jumlah kecil dan masih tergolong wajar. Sebanyak 4 gambar *Crowntail* diklasifikasikan sebagai *Halfmoon*, kemungkinan disebabkan oleh fitur visual tertentu seperti bentuk ekor atau orientasi tubuh yang menyerupai karakteristik *Halfmoon*. Selain itu, terdapat 6 gambar *Halfmoon* yang diklasifikasikan sebagai *Plakat*, dan 5 gambar *Plakat* yang diklasifikasikan sebagai *Halfmoon*. Kesalahan-kesalahan ini bisa disebabkan oleh faktor-faktor seperti kemiripan visual antar kelas, kualitas gambar yang kurang baik, atau sudut pengambilan gambar yang tidak optimal.

Meskipun terdapat beberapa *misclassifications*, jumlahnya relatif sangat kecil jika dibandingkan dengan jumlah prediksi yang benar. Hal ini menunjukkan bahwa model *Vision Transformer* masih sangat andal dalam mengklasifikasikan jenis ikan cupang meskipun proporsi data pelatihan dikurangi. Model ini tetap mampu menangkap fitur-fitur penting dari masing-masing kelas dan melakukan prediksi dengan akurasi yang tinggi pada data uji.

### 4. Pengujian dengan Rasio 50:50

Pengujian keempat dilakukan dengan membagi *Dataset* secara seimbang, yaitu 50% untuk pelatihan dan 50% untuk pengujian. Dari total 1.203 citra ikan cupang, sebanyak 601 digunakan sebagai data latih dan 602 sebagai data uji. Arsitektur model tetap menggunakan *vit\_base\_patch16\_224* yang telah dimodifikasi untuk klasifikasi tiga kelas ikan cupang: *Crowntail*, *Halfmoon*, dan *Plakat*.

Pelatihan dilakukan selama 15 *epoch* menggunakan *GPU*, dan model terbaik tersimpan otomatis pada *epoch* ke-6 dengan akurasi validasi sebesar 98,17% dan loss sebesar 0.0568. Meskipun data pelatihan lebih sedikit dibandingkan skenario sebelumnya, performa validasi justru menunjukkan hasil yang sangat tinggi dan stabil.

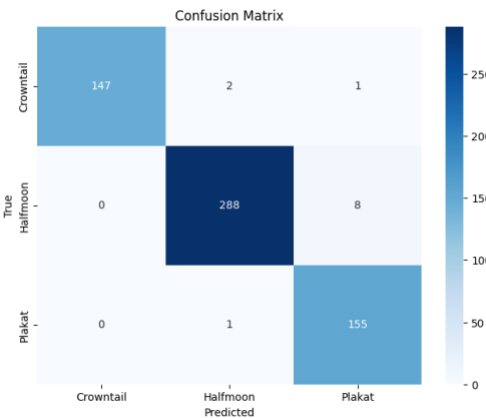
Pada akhir pelatihan, akurasi pelatihan mencapai 100%, menandakan bahwa model telah mempelajari pola dengan sangat baik

dari data latih. Evaluasi terhadap data uji memperlihatkan hasil klasifikasi yang sangat akurat, sebagaimana dirangkum dalam Tabel 5.

Tabel 5. *Classification Report* Pengujian Rasio 50:50

| Kelas            | Precision | Recall | F1-Score | Support |
|------------------|-----------|--------|----------|---------|
| <i>Crowntail</i> | 1.00      | 0.98   | 0.98     | 150     |
| <i>Halfmoon</i>  | 0.99      | 0.97   | 0.98     | 296     |
| <i>Plakat</i>    | 0.95      | 0.99   | 0.97     | 156     |
| Accuracy         | —         | —      | 0.98     | 602     |
| Macro Avg        | 0.98      | 0.98   | 0.98     | 602     |
| Weighted Avg     | 0.98      | 0.98   | 0.98     | 602     |

Visualisasi hasil pengujian dalam bentuk *Confusion Matrix* disajikan pada Gambar 6.



Gambar 6. *Confusion Matrix* Pengujian Rasio 50:50

Berdasarkan hasil evaluasi dengan rasio data latih dan uji sebesar 50:50, model *Vision Transformer* masih menunjukkan performa klasifikasi yang sangat baik, meskipun terjadi sedikit peningkatan jumlah kesalahan klasifikasi dibandingkan rasio sebelumnya. Model mampu mengklasifikasikan 147 gambar *Crowntail*, 288 gambar *Halfmoon*, dan 155 gambar *Plakat* dengan benar. Angka ini menunjukkan bahwa mayoritas prediksi dilakukan secara akurat, bahkan ketika jumlah data latih dikurangi secara signifikan.

Namun, beberapa kesalahan klasifikasi tetap ditemukan. Sebanyak 8 gambar *Halfmoon* teridentifikasi secara keliru sebagai *Plakat*, yang mungkin disebabkan oleh kemiripan pola sirip atau warna pada gambar tertentu. Selain itu, 2 gambar *Crowntail* terklasifikasi sebagai *Halfmoon*, 1 gambar *Crowntail* teridentifikasi sebagai *Plakat*, dan 1 gambar *Plakat* diklasifikasikan sebagai *Halfmoon*. Kesalahan ini kemungkinan besar terjadi akibat kemiripan visual antar kelas atau karena variasi sudut pengambilan gambar dan pencahayaan yang dapat mempengaruhi fitur visual yang dikenali oleh model.

Secara umum, meskipun jumlah data pelatihan dikurangi menjadi separuh, model tetap dapat melakukan klasifikasi dengan tingkat akurasi yang tinggi. Hal ini menunjukkan bahwa *Vision Transformer* memiliki kemampuan generalisasi yang kuat dan tetap mampu membedakan fitur-fitur khas dari masing-masing jenis ikan cupang secara efektif, bahkan dalam kondisi pelatihan yang lebih terbatas.

##### 5. Perbandingan Hasil Pelatihan pada Berbagai Rasio Data

Pada subbab ini dilakukan perbandingan terhadap hasil pengujian model *Vision Transformer* pada empat skenario rasio pembagian data pelatihan dan data pengujian, yaitu 80:20, 70:30, 60:40, dan 50:50. Tujuan dari perbandingan ini adalah untuk mengevaluasi dampak perubahan proporsi data terhadap performa model dalam mengklasifikasikan jenis ikan cupang.

Tabel 6 merangkum hasil akurasi dan nilai rata-rata dari metrik *Precision*, *Recall*, dan *F1-Score* pada masing-masing rasio.

Tabel 6. Hasil Evaluasi Model pada Berbagai Rasio Pengembangan Data

| Rasio Data | Akurasi | Precision (avg) | Recall (avg) | F1-score (avg) |
|------------|---------|-----------------|--------------|----------------|
| 80:20      | 0.98    | 0.98            | 0.98         | 0.98           |
| 70:30      | 0.98    | 0.98            | 0.98         | 0.98           |
| 60:40      | 0.97    | 0.97            | 0.97         | 0.97           |
| 50:50      | 0.98    | 0.98            | 0.98         | 0.98           |

Dari hasil pada Tabel 6, terlihat bahwa model *Vision Transformer* pada berbagai rasio pembagian data latih dan data uji, dapat dilihat bahwa model menunjukkan performa yang sangat baik di semua rasio yang diuji. Namun, terdapat penurunan kecil pada nilai akurasi dan metrik lainnya seiring dengan berkurangnya proporsi data latih.

Secara spesifik, rasio 80:20 dan 70:30 menghasilkan akurasi dan *f1-score* tertinggi, yaitu sekitar 98–99%, yang menunjukkan bahwa dengan jumlah data latih yang memadai, model mampu mempelajari pola klasifikasi jenis ikan cupang secara efektif. Pada rasio 60:40, terjadi sedikit penurunan performa dengan akurasi mencapai 97%, dan pada rasio 50:50 akurasi menurun lebih lanjut menjadi sekitar 97%, meskipun performa ini masih tergolong sangat baik untuk tugas klasifikasi citra.

Penurunan performa ini mengindikasikan bahwa semakin besar proporsi data latih yang digunakan, semakin baik kemampuan model dalam mengenali fitur-fitur penting dari setiap kelas ikan cupang. Sebaliknya, ketika jumlah data latih berkurang, model cenderung kurang optimal dalam menggeneralisasi terhadap data uji, sehingga akurasi sedikit menurun. Dengan demikian, rasio pembagian data latih dan uji yang ideal perlu mempertimbangkan keseimbangan antara ketersediaan data latih yang cukup dan data uji yang representatif untuk evaluasi.

##### 3.2. Pengujian Model dengan Variasi Rasio Data Latih dan Data Uji

Pada tahap ini, model *Vision Transformer* diuji menggunakan empat variasi rasio pembagian data latih dan data uji, yaitu 80:20, 70:30, 60:40, dan 50:50. Tujuan dari pengujian ini adalah untuk mengevaluasi seberapa besar pengaruh proporsi data latih terhadap performa model dalam melakukan klasifikasi citra ikan cupang ke dalam tiga kelas, yaitu *Crowntail*, *Halfmoon*, dan *Plakat*.

Masing-masing skenario pengujian dilakukan dengan memantau metrik evaluasi seperti akurasi, *Precision*, *Recall*, dan *F1-Score*. Selain itu, untuk menguji kemampuan generalisasi model terhadap data baru, dilakukan juga pengujian tambahan menggunakan tiga gambar uji dari setiap kelas yang tidak pernah dilihat sebelumnya oleh model.

Hasil pengujian pada setiap rasio disajikan secara terpisah pada subbab berikut, disertai dengan analisis performa model berdasarkan prediksi probabilitas dan ketepatan klasifikasi. Pada akhir bagian ini, disusun kesimpulan perbandingan untuk memberikan gambaran umum terhadap kecenderungan performa model pada tiap rasio.

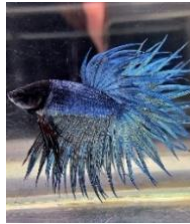
Untuk pengujian model *Vision Transformer*, digunakan tiga sampel gambar ikan cupang yang mewakili kelas *Crowntail*, *Halfmoon*, dan *Plakat*. Gambar ikan diambil langsung menggunakan data asli yaitu menggunakan ikan yang dimiliki sendiri. Untuk gambar *Tidak Dikenali* gambar diambil dari *Instagram @ikanguppy.oke*. Pemilihan gambar ini bertujuan untuk merepresentasikan variasi *visual* yang sesungguhnya pada masing-masing jenis ikan cupang sehingga dapat menguji kemampuan model dalam mengklasifikasikan jenis ikan secara akurat dan efektif pada data pengujian.



Gambar 7. Sampel Gambar *Halfmoon*



Gambar 8. Sampel Gambar *Plakat*



Gambar 9. Sample Gambar *Crowntail*

### 1. Hasil Pengujian pada Rasio 80:20

Pengujian model *Vision Transformer* dengan rasio pembagian data pelatihan dan data pengujian sebesar 80:20 menunjukkan performa yang sangat optimal. Model berhasil mencapai akurasi sebesar 99%, dengan nilai rata-rata *precision*, *recall*, dan *f1-score* masing-masing sebesar 0,99. Capaian ini mengindikasikan bahwa model memiliki kapabilitas yang sangat baik dalam mengenali serta mengklasifikasikan citra ikan cupang ke dalam tiga kelas yang tersedia, yaitu *Crowntail*, *Halfmoon*, dan *Plakat*.

Sebagai bentuk validasi tambahan terhadap generalisasi model, dilakukan pengujian menggunakan tiga citra tunggal yang merepresentasikan masing-masing kelas. Hasil prediksi menunjukkan bahwa seluruh gambar berhasil diklasifikasikan secara akurat dan sesuai dengan kelas aslinya. Gambar dari kelas *Crowntail* diprediksi dengan probabilitas 1.0000, gambar *Halfmoon* dengan probabilitas 0.9998, dan gambar *Plakat* dengan probabilitas 0.9999. Tingginya nilai probabilitas ini menandakan tingkat keyakinan model yang sangat kuat terhadap keputusan klasifikasinya.

Dengan demikian, rasio pembagian data sebesar 80:20 terbukti mampu menghasilkan model dengan performa tinggi dan tingkat keandalan yang sangat baik dalam mengidentifikasi citra ikan cupang. Rasio ini dapat dijadikan acuan ideal untuk pengembangan sistem klasifikasi citra serupa dalam konteks aplikasi nyata.



Gambar 10. Hasil Prediksi Rasio 80:20

### 2. Hasil Pengujian pada Rasio 70:30

Pada pengujian dengan rasio pembagian data pelatihan dan data pengujian sebesar 70:30, model *Vision Transformer* tetap menunjukkan performa klasifikasi yang sangat baik. Model berhasil mencapai akurasi sebesar 99%, disertai dengan nilai rata-rata *precision*, *recall*, dan *f1-score* sebesar 0,99. Capaian ini menunjukkan bahwa penurunan proporsi data pelatihan sebesar 10% dari rasio sebelumnya tidak secara signifikan menurunkan kemampuan model dalam mengenali dan mengklasifikasikan citra ikan cupang ke dalam kelas yang tepat.

Sebagai bagian dari evaluasi lebih lanjut, tiga gambar uji dari masing-masing kelas ikan cupang diuji secara individual. Seluruh gambar berhasil diklasifikasikan dengan benar dan tingkat keyakinan

yang sangat tinggi. Gambar *Crowntail* diprediksi dengan probabilitas sempurna sebesar 1.0000, gambar *Halfmoon* dengan probabilitas 0.9964, dan gambar *Plakat* dengan probabilitas 0.9998. Hasil ini menunjukkan bahwa meskipun jumlah data pelatihan lebih sedikit dibandingkan rasio 80:20, model tetap mampu menghasilkan prediksi yang akurat dan konsisten.

Dengan demikian, rasio 70:30 dapat dianggap tetap efektif dalam melatih model untuk mencapai generalisasi yang sangat baik terhadap data yang belum pernah dilihat sebelumnya, menjadikannya alternatif yang dapat diandalkan dalam skenario klasifikasi citra nyata.



Gambar 11. Hasil Prediksi Rasio 70:30

### 3. Hasil Pengujian pada Rasio 60:40

Pada skenario pengujian dengan rasio pembagian data pelatihan dan pengujian sebesar 60:40, model *Vision Transformer* tetap menunjukkan performa klasifikasi yang sangat baik. Model mencatat akurasi sebesar 98%, dengan nilai rata-rata *precision*, *recall*, dan *f1-score* sebesar 0,98. Meskipun terdapat sedikit penurunan dibandingkan rasio 70:30 dan 80:20, hasil ini tetap mencerminkan kemampuan model yang tinggi dalam mengklasifikasikan citra ikan cupang ke dalam tiga kelas yang tersedia.

Sebagai bagian dari pengujian lebih lanjut, tiga gambar uji dari masing-masing kelas diuji secara individual. Seluruh gambar berhasil dikenali dengan benar sesuai kelas aslinya. Gambar *Crowntail* diprediksi dengan probabilitas 0.9997, *Halfmoon* dengan probabilitas sempurna 1.0000, dan *Plakat* dengan probabilitas 0.9987. Tingkat probabilitas yang sangat tinggi menunjukkan bahwa model masih memiliki keyakinan yang kuat terhadap prediksi yang dihasilkan, meskipun data pelatihan yang digunakan lebih sedikit.

Hasil ini menandakan bahwa meskipun terjadi penurunan akurasi secara agregat, kemampuan generalisasi model terhadap data baru tetap terjaga dengan baik pada rasio 60:40. Namun demikian, jika mempertimbangkan penerapan dalam skenario nyata yang menuntut akurasi tinggi secara konsisten, disarankan untuk tetap menggunakan rasio data pelatihan yang lebih besar guna memastikan stabilitas dan keandalan model dalam klasifikasi.



Gambar 12. Hasil Prediksi Rasio 60:40

### 4. Hasil Pengujian pada Rasio 50:50

Pada skenario pengujian dengan rasio pembagian data sebesar 50:50 antara data latih dan data uji, model tetap mampu menunjukkan performa yang sangat baik meskipun jumlah data latih berkurang secara signifikan. Model mencapai nilai akurasi sebesar 97%, dengan nilai rata-rata *precision*, *recall*, dan *f1-score* masing-masing sebesar 0,97. Hal ini mengindikasikan bahwa model masih memiliki kemampuan generalisasi yang kuat terhadap data yang belum pernah dilihat sebelumnya, meskipun dilatih dengan jumlah data yang lebih terbatas.

Hasil evaluasi pada citra uji menunjukkan bahwa ketiga gambar sampel yang mewakili kelas *Crowntail*, *Halfmoon*, dan *Plakat* berhasil diklasifikasikan secara akurat oleh model. Prediksi terhadap gambar

*Crowntail* menghasilkan probabilitas sebesar 1.0000, sementara gambar *Halfmoon* dan *Plakat* masing-masing diprediksi dengan probabilitas sebesar 0.9999 dan 0.9980. Tingginya nilai probabilitas ini menunjukkan tingkat kepercayaan model yang tinggi terhadap hasil klasifikasi yang diberikan.

Meskipun demikian, dibandingkan dengan rasio data latih yang lebih besar, performa model pada rasio 50:50 mengalami sedikit penurunan secara keseluruhan. Fenomena ini mendukung temuan sebelumnya bahwa pengurangan jumlah data pelatihan dapat berdampak terhadap penurunan kemampuan model dalam mengekstraksi fitur penting dari setiap kelas secara konsisten. Oleh karena itu, dapat disimpulkan bahwa penggunaan rasio data latih yang lebih besar tetap lebih disarankan guna menghasilkan model yang lebih stabil dan akurat dalam menghadapi data nyata.



Gambar 13. Hasil Prediksi Rasio 50:50

#### 5. Kesimpulan Perbandingan Pengujian

Berdasarkan hasil pengujian terhadap empat skenario rasio pembagian data pelatihan dan pengujian (80:20, 70:30, 60:40, dan 50:50), dapat disimpulkan bahwa performa model *Vision Transformer* dalam mengklasifikasikan jenis ikan cupang menunjukkan kecenderungan menurun seiring dengan berkurangnya proporsi data pelatihan. Pada rasio 80:20 dan 70:30, model mencapai performa terbaik dengan akurasi sebesar 99% serta keberhasilan klasifikasi yang sempurna terhadap gambar uji dari ketiga kelas (*Crowntail*, *Halfmoon*, dan *Plakat*), masing-masing dengan probabilitas prediksi di atas 0,996.

Pada rasio 60:40, meskipun akurasi menurun menjadi 98%, model masih mampu mengklasifikasikan ketiga gambar uji dengan benar dan probabilitas prediksi yang sangat tinggi, menunjukkan kemampuan generalisasi yang masih kuat. Namun, penurunan performa mulai terlihat secara lebih signifikan pada rasio 50:50, dengan akurasi turun menjadi 97%. Meskipun model tetap mampu mengklasifikasikan gambar uji dengan benar, akurasi agregat dan metrik evaluasi lainnya mengindikasikan adanya penurunan stabilitas prediksi.

Temuan ini menegaskan bahwa model sangat bergantung pada ketersediaan data pelatihan yang mencukupi untuk mempelajari fitur visual yang membedakan setiap kelas secara optimal. Selain itu, sensitivitas model terhadap kelas yang memiliki kemiripan visual, seperti *Plakat* dan *Halfmoon*, menjadi lebih nyata ketika jumlah data latih dikurangi. Oleh karena itu, pemilihan rasio pembagian data pelatihan dan pengujian merupakan aspek krusial dalam proses pengembangan sistem klasifikasi citra berbasis *deep learning*, guna memastikan model mampu melakukan generalisasi dengan baik terhadap data yang belum pernah dilihat sebelumnya.

#### 4. SIMPULAN

Penelitian ini bertujuan untuk membangun sistem klasifikasi citra guna mengidentifikasi jenis ikan cupang menggunakan arsitektur *Vision Transformer*. Model dilatih dan diuji dengan menggunakan *Dataset* yang mencakup tiga kelas utama ikan cupang, yaitu *Crowntail*, *Halfmoon*, dan *Plakat*. Proses pelatihan dilakukan dengan membandingkan performa model pada empat rasio pembagian data pelatihan dan pengujian, yaitu 80:20, 70:30, 60:40, dan 50:50.

Penelitian ini dilakukan untuk mengembangkan sistem klasifikasi citra menggunakan arsitektur *Vision Transformer* guna mengidentifikasi jenis ikan cupang secara otomatis. Sistem ini

dirancang agar mampu memberikan kemudahan bagi penghobi atau peminat ikan cupang dalam mengenali jenis ikan berdasarkan standar kontes.

Berdasarkan hasil penelitian dan analisis yang telah dilakukan, dapat disimpulkan bahwa:

1. Model klasifikasi citra berbasis *Vision Transformer* yang dikembangkan dalam penelitian ini berhasil mengidentifikasi jenis ikan cupang secara otomatis dengan tingkat akurasi tinggi dan performa yang baik pada berbagai rasio pembagian data pelatihan dan pengujian. Hal ini menunjukkan bahwa *Vision Transformer* efektif digunakan dalam klasifikasi jenis ikan cupang berdasarkan citra.
2. Sistem yang dikembangkan dapat memberikan kemudahan bagi para penghobi maupun peminat ikan cupang dalam mengenali dan menentukan jenis ikan cupang yang sesuai dengan kategori umum kontes. Dengan demikian, model ini berpotensi menjadi alat bantu yang berguna dalam mendukung kegiatan seleksi dan klasifikasi ikan cupang secara cepat dan akurat

#### DAFTAR PUSTAKA

- [1] Arshal, M. (2023). 2023 Muh Arshal 4516033013 (2). Universitas Bosowa.
- [2] Baso, B., & Suciati, N. (2020). *Temu Kembali Citra Tenun Nusa Tenggara Timur Menggunakan Ekstraksi Fitur Yang Robust Terhadap Perubahan Skala, Rotasi Dan Pencahayaannya*. <https://doi.org/10.25126/jtiik.202072002>
- [3] Bai, Y., Chen, M., Zhou, P., Zhao, T., Lee, J. D., Kakade, S., Wang, H., & Xiong, C. (2020). *How Important is the Train-Validation Split in Meta-Learning?* <http://arxiv.org/abs/2010.05843>
- [4] Boesch, G. (2023, November 25). *Vision Transformers (ViT) in Image Recognition: Full Guide - viso.ai*. <https://viso.ai/deeplearning/Vision-Transformer-vit/>
- [5] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. <http://arxiv.org/abs/2010.11929>
- [6] Goodfellow, I., Bengio, Y., & Courville, A. (n.d.). *Deep Learning*. Retrieved June 9, 2025, from <http://www.deeplearningbook.org>
- [7] Hasym, I. E., & Susilawati, I. (2021). *KONSTELASI: Konvergensi Teknologi dan Sistem Informasi Klasifikasi Jenis Ikan Cupang Menggunakan Algoritma Principal Component Analysis (PCA) Dan K-Nearest Neighbors (KNN)*.
- [8] Kashyap, P. (2024, December 1). *How to Save and Load Checkpoints for Training a CNN with PyTorch* | by Piyush Kashyap | Medium. <https://medium.com/%40piyushkashyap045/how-to-save-and-load-checkpoints-for-training-a-cnn-with-pytorch-e17395cddb3d>
- [9] Melisah. (2023). *Klasifikasi Rempah Menggunakan Ekstraksi Fitur Histogram Of Oriented Gradient (Hog) Dengan Metode K-Nearest Neighbor (K-Nn) [Universitas Medan Area]*. <https://repositori.uma.ac.id/handle/123456789/22682>
- [10] Putra, F. A.-I. A., Sulaksono, A. G., Utomo, L. T., & Khamdani, A. R. (2023). *Klasifikasi Buah dan Sayur Menggunakan Fitur Ekstraksi HOG dan Metode KNN*. *Jurnal Informatika Polinema*, 10(1). <https://doi.org/10.33795/JIP.V10I1.1433>
- [11] Quaranta, L., Calefato, F., & Lanubile, F. (2022). *Eliciting Best Practices for Collaboration with Computational Notebooks. Proceedings of the ACM on Human-Computer Interaction*, 6(CSCWI). <https://doi.org/10.1145/3512934>
- [12] Rinata, A. R., Ghofur, M. A., & Widodo, H. P. (2023). *Jurnal Komunikasi Nusantara Implementasi Media Digital Marketing*

- Communications Pada Pedagang Ikan Cupang Hias*. Jurnal Komunikasi Nusantara, 5(2), 273–283. <https://doi.org/10.33366/jkn.v%vi%i.417>
- [13] Rizal, F. (2021, July 1). *5 Jenis Ikan Cupang yang Berbentuk Cantik*. <https://www.halodoc.com/artikel/5-jenis-ikan-cupang-yang-berbentuk-cantik>
- [14] Saputra, G. A., & Agastya, I. M. A. (2024). *Betta Fish Identification System Based On Convolutional Neural Network*. Journal of Applied Informatics and Computing, 8(2), 443–452. <https://doi.org/10.30871/JAIC.V8I2.8449>
- [15] Sarasuartha, M. N., Desiana, P., Ayu, W., & Huizen, R. R. (2024). *Pengaruh Optimizer Adam, AdamW, SGD, dan LAMB terhadap Model Vision Transformer pada Klasifikasi Penyakit Paru-paru*. Prosiding Seminar Hasil Penelitian Informatika Dan Komputer 2024, Vol. 1 No. 22024. <https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database>
- [16] Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2020). *Training data-efficient image Transformers & distillation through attention*. Proceedings of Machine Learning Research, 139, 10347–10357. <https://arxiv.org/pdf/2012.12877>
- [17] Uthama, R., Yuhandri, & Billy Hendrik. (2024). *Vision Transformer untuk Identifikasi 15 Variasi Citra Ikan Koi*. Jurnal CoSciTech (Computer Science and Information Technology), 5(1), 159–168. <https://doi.org/10.37859/coscitech.v5i1.6711>
- [18] Wang, Z., Wang, P., Liu, K., Wang, P., Fu, Y., Lu, C.-T., Aggarwal, C. C., Pei, J., & Zhou, Y. (2024). *A Comprehensive Survey on Data Augmentation*. <http://arxiv.org/abs/2405.09591>
- [19] Wulaning Ayu, P. D., & Pradipta, G. A. (2022). *U-Net Tuning Hyperparameter for Segmentation in Amniotic Fluid Ultrasonography Image*. 2022 4th International Conference on Cybernetics and Intelligent System, ICORIS 2022. <https://doi.org/10.1109/ICORIS56080.2022.10031294>
- [20] Yang, C., Zhou, P., Wang, C. S., Fu, G. Y., Xu, X. W., Niu, Z., Zhu, L., Yuan, Y., Shen, H. Bin, & Pan, X. (2024). *FishAI: Automated hierarchical marine fish image classification with Vision Transformer*. Engineering Reports. <https://doi.org/10.1002/eng2.12992>