TUGAS EKSPLORASI

PERBANDINGAN ViT TINY DAN SWIN TRANSFORMER (STUDI KASUS: KLASIFIKASI GAMBAR MENGGUNAKAN DATASET CIFAR-10)

Disusun Oleh:

Arkan Hariz Chandrawinata Liem
NIM: 122140038

Mata Kuliah:
Deep Learning

Dosen Pengampu:
Imam Eko Wicaksono, S.Si., M.Si.
Martin Clinton Tosima Manullang, S.T., M.T., Ph.D.

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
INSTITUT TEKNOLOGI SUMATERA
2025

GitHub Repository

https://github.com/ArkanHariz/VisionTransformer-Comparison

Contents

1 Pendahuluan

1.1 Latar Belakang

Perkembangan teknologi *computer vision* telah terjadi dengan sangat cepat dalam sepuluh tahun terakhir, terutama disebabkan oleh munculnya teknik *Deep Learning*. Arsitektur *Convolutional Neural Networks* (CNN) telah menjadi standar dan teknik yang paling populer untuk menyelesaikan berbagai tugas pemrosesan citra. Tahun 2020 dimulai dengan kemenangan CNN dalam mengekstraksi fitur visual lokal. Namun, dominasi arsitektur konvensional ini mulai menghadapi tantangan besar dari pendekatan baru berbasis *Transformer* seiring dengan meningkatnya kompleksitas data dan kebutuhan akan pemahaman konteks global yang lebih baik [1].

Pergeseran paradigma yang signifikan kemudian terjadi dengan diperkenalkannya arsitektur Vision Transformer (ViT). Studi ini memberikan bukti faktual yang mendukung gagasan bahwa ketergantungan pada lapisan konvolusi (CNN) tidak lagi diperlukan untuk pengenalan citra. ViT menggunakan pendekatan yang berbeda secara radikal dengan memecah gambar menjadi urutan patch (potongan kecil) berukuran tetap dan memprosesnya menggunakan mekanisme self-attention biasa, mirip dengan cara model bahasa memproses urutan kata [2]. Ketika dilatih dengan dataset yang cukup besar, metode ini memungkinkan model untuk memiliki efisiensi komputasi yang lebih baik pada skala besar dan mencapai akurasi yang setara atau bahkan melampaui model CNN yang modern.

Keunggulan fundamental *Vision Transformer* terletak pada kemampuannya dalam memahami konteks gambar secara menyeluruh. Terdapat perbedaan antara *Vision Transformer* dengan CNN. CNN cenderung fokus pada fitur lokal (seperti tekstur dan tepian) pada lapisan-lapisan awal sedangkan ViT terbukti mampu menangkap hubungan jarak jauh (*long-range dependencies*) dan informasi spasial global sejak lapisan pertama [3]. Mekanisme ini memungkinkan model untuk menghubungkan bagian-bagian gambar yang berjauhan, misalnya antara kepala dan kaki subjek—secara langsung tanpa harus melalui tahapan hirarkis yang panjang, menghasilkan representasi fitur yang lebih kaya dan robust terhadap gangguan latar belakang.

1.2 Motivasi Perbandingan Model

Eksperimen ini dilakukan untuk membandingkan dua arsitektur *Vision Transformer* yang mewakili pendekatan berbeda dalam pemrosesan citra, dengan motivasi sebagai berikut:

1. **Global vs. Hierarkis**: Membandingkan efektivitas mekanisme Global Self-Attention pada ViT melawan pendekatan Hierarchical Shifted Windows pada Swin Transformer dalam mengenali fitur citra.
2. **Efisiensi Komputasi**: Menguji secara empiris apakah kompleksitas linear pada Swin mampu memberikan kecepatan inferensi (throughput) yang lebih unggul dibandingkan kompleksitas kuadratik pada ViT.
3. **Trade-off Arsitektur Tiny**: Menganalisis apakah kompleksitas pada arsitektur Swin sebanding dengan peningkatan akurasi yang dihasilkan dibandingkan varian ViT-Tiny yang lebih sederhana.

1.3 Tujuan Eksperimen

Tujuan dari eksperimen ini adalah:

1. Memahami perbedaan arsitektur antara Vision Transformer (ViT) dan Swin Transformer.
2. Mengimplementasikan dan melatih minimal 2 model Vision Transformer.
3. Melakukan analisis berdasarkan metrik performa, jumlah parameter, dan waktu inferensi.

2 Landasan Teori

2.1 Transformer dan Self-Attention

Arsitektur *Transformer* pertama kali diperkenalkan dalam penelitian yang berjudul "*Attention Is All You Need*" [4]. Arsitektur ini, pertama kali dirancang untuk tugas pemrosesan bahasa alami (*Natural Language Processing*, atau NLP), membawa perubahan besar ke dalam bidang pembelajaran mendalam. Dengan menggunakan mekanisme rekurensi (seperti RNN atau LSTM) dan konvolusi (CNN), dan sepenuhnya bergantung pada mekanisme atensi (*attention mechanism*) untuk menangkap ketergantungan global antara input dan output. Mekanisme Self- Attention adalah komponen inti dari Transformer adalah Scaled Dot-Product Attention. Dengan bantuan mekanisme ini, model dapat menghubungkan posisi yang berbeda dari satu urutan untuk menghitung representasinya. Self- attention memungkinkan setiap elemen input untuk "memperhatikan" atau berinteraksi dengan seluruh elemen lain secara langsung, terlepas dari jarak posisinya, berbeda dengan CNN yang memproses informasi secara lokal. Ini membuat Transformer sangat baik untuk menangkap hubungan jarak jauh.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

Secara matematis, fungsi atensi dipetakan melalui query (Q), key (K), dan value (V). Output dihitung sebagai jumlah bobot dari value, di mana bobot tersebut ditentukan oleh kecocokan antara query dengan key yang bersesuaian. Di mana $\sqrt{d_k}$ adalah faktor penskalaan untuk mencegah hasil dot product menjadi terlalu besar yang dapat menyebabkan gradien menghilang pada fungsi softmax.

2.2 Vision Transformer (ViT)

Vision Transformer (ViT) adalah upaya untuk menerapkan arsitektur Transformer standar pada gambar dengan sedikit modifikasi. ViT membagi gambar menjadi patch (potongan) berukuran tetap, berbeda dengan CNN yang memproses piksel. Selanjutnya, setiap patch dihaluskan (diratakan) menjadi vektor satu dimensi dan diproyeksikan ke dalam ruang embedding linear. Keunggulan utama ViT adalah kemampuannya dalam menangkap konteks global gambar sejak lapisan awal, namun model ini membutuhkan dataset pre-training yang sangat besar [2].

2.3 Swin Transformer

Swin Transformer dibuat dengan tujuan untuk mengatasi masalah komputasi pada ViT standar, terutama ketika menangani gambar resolusi tinggi. Swin memperkenalkan struktur hierarkis yang menyerupai CNN, di mana representasi gambar dimulai pada ukuran yang lebih kecil dan secara bertahap digabungkan pada lapisan yang lebih dalam melalui proses yang dikenal sebagai patch merging. Mekanisme Jendela Bergeser, atau Jendela Bergeser, adalah inovasi utama Swin Transformer. Alih-alih menghitung self-attention secara global (untuk semua patch), Swin membatasi perhitungan atensi hanya pada jendela lokal yang tidak saling tumpang tindih. Partisi jendela digeser ke lapisan berikutnya untuk memungkinkan jendela berkomunikasi satu sama lain. Metode ini menghasilkan kompleksitas komputasi linear terhadap ukuran gambar ($O(N)$) dengan tingkat efisiensi yang lebih tinggi dibandingkan ViT konvensional yang memiliki kompleksitas kuadratik ($O(N^2)$), yang membuat Swin sangat efektif untuk berbagai tugas, seperti deteksi objek dan segmentasi [5].

2.4 Perbedaan Kunci Antar Model

Berikut adalah perbandingan ViT dengan Swin Transformer

Tabel 1: Perbandingan Vision Transformer (ViT) vs Swin Transformer

Aspek	Vision Transformer (ViT)	Swin Transformer
Struktur Arsitektur	Isotropic (Resolusi tetap)	Hierarchical (Multi-scale)
Attention Mechanism	Global Self-Attention	Local (Shifted Windows)
Kompleksitas	Kuadratik $O(N^2)$	Linear $O(N)$
Positional Emb.	Absolute	Relative Bias
Patch Size Awal	16×16	4×4
Scope Interaksi	Global (Seluruh gambar)	Lokal (Dalam window)
Best Use Case	Klasifikasi Gambar	Dense Prediction (Seg/Det)

2.5 Kelebihan dan Kekurangan Masing-Masing Model

Berikut adalah analisis kelebihan dan kekurangan dari Vision Transformer (ViT) dan Swin Transformer.

1. Vision Transformer (ViT)

- **Kelebihan:**
  - **Global Receptive Field:** Mampu menangkap hubungan jarak jauh antar *patch* gambar sejak lapisan pertama, memberikan pemahaman konteks semantik yang utuh.
  - **Skalabilitas:** Performa model cenderung terus meningkat secara signifikan ketika dilatih pada dataset berskala sangat besar (seperti JFT-300M).
- **Kekurangan:**
  - **Kompleksitas Kuadratik:** Biaya komputasi memori meningkat secara kuadratik $O(N^2)$ terhadap resolusi gambar, sehingga kurang efisien untuk gambar beresolusi tinggi.
  - **Data Hungry:** Karena minimnya *inductive bias* (seperti pada CNN), ViT membutuhkan jumlah data pelatihan yang sangat besar atau teknik augmentasi yang kuat untuk menghindari *overfitting*.

2. Swin Transformer

- **Kelebihan:**
  - **Efisiensi Linear:** Menggunakan mekanisme *Window-based Attention* yang menghasilkan kompleksitas komputasi linear $O(N)$, memungkinkan pemrosesan gambar resolusi tinggi dengan lebih cepat.
  - **Struktur Hierarkis:** Menghasilkan fitur *multi-scale* yang sangat efektif untuk tugas-tugas *dense prediction* seperti deteksi objek dan segmentasi citra.
- **Kekurangan:**
  - **Kompleksitas Implementasi:** Mekanisme *Shifted Windows* menambah kerumitan arsitektur dibandingkan dengan *self-attention* standar.
  - **Konteks Global Tertunda:** Informasi global baru terbentuk secara bertahap melalui penggabungan jendela pada lapisan-lapisan yang lebih dalam, berbeda dengan ViT yang langsung global sejak awal.

3 Metodologi

3.1 Deskripsi Dataset

Eksperimen ini menggunakan dataset CIFAR-10 dari Canadian Institute for Advanced Research, yang merupakan salah satu tolok ukur standar yang paling umum digunakan untuk mengevaluasi algoritma pengenalan citra. CIFAR-10 terdiri dari 60.000 gambar berwarna dengan resolusi rendah yang dibagi secara merata ke dalam sepuluh kelas yang saling eksklusif. Airplane, car, bird, cat, deer, dog, frog, horse, ship, dan truck adalah sepuluh kategori tersebut. Setiap kelas memiliki 6.000 gambar. Standar resmi mengatur pembagian data: 50.000 foto digunakan sebagai set pelatihan (training set) dan 10.000 foto digunakan sebagai set uji (test set).

Tabel 2: Deskripsi Statistik Dataset CIFAR-10

Atribut	**Keterangan**
Nama Dataset	CIFAR-10
Jumlah Kelas	10 Kelas
Resolusi Asli	32×32 piksel
Saluran Warna	3 (RGB)
Total Data	60.000 Citra
Data Training	50.000 Citra
Data Testing	10.000 Citra
Target Resize (ViT/Swin)	224×224 piksel

3.2 Preprocessing dan Augmentasi Data

Berdasarkan implementasi kode, proses *preprocessing* dilakukan untuk menyesuaikan dimensi dan distribusi data CIFAR-10 agar kompatibel dengan arsitektur model *pre-trained*. Tahapan yang dilakukan adalah sebagai berikut:

1. **Resizing:** Mengubah resolusi citra dari ukuran asli 32×32 piksel menjadi 224×224 piksel. Langkah ini sangat penting karena arsitektur ViT dan Swin Transformer yang digunakan membutuhkan input dimensi tinggi untuk membagi *patch* secara efektif.
2. **Augmentasi Data:** Pada data latih (*training set*), diterapkan transformasi `RandomHorizontalFlip` untuk membalik citra secara horizontal dengan probabilitas 50%. Ini bertujuan memperkaya variasi data dan mengurangi risiko *overfitting*.
3. **Normalisasi:** Citra dinormalisasi menggunakan nilai *mean* dan *standard deviation* dari dataset ImageNet:
   - Mean: $[0.485, 0.456, 0.406]$
   - Std: $[0.229, 0.224, 0.225]$

   Normalisasi ini memastikan input memiliki distribusi yang serupa dengan data yang digunakan saat *pre-training* model, sehingga mempercepat konvergensi saat proses *fine-tuning*.

3.3 Konfigurasi Training

Kedua model dilatih dengan hyperparameter yang seragam. Proses pelatihan dilakukan selama 10 epoch menggunakan ukuran batch sebesar 16. Ukuran batch ini dipilih menyesuaikan dengan kapasitas memori GPU yang tersedia. Optimasi bobot model dilakukan menggunakan AdamW (Adam with Weight Decay), yang merupakan standar de-facto untuk pelatihan model berbasis Transformer. Learning rate ditetapkan

sebesar 5e-5 untuk kedua model guna menjaga stabilitas gradien selama proses fine-tuning. Fungsi kerugian (loss function) yang digunakan adalah Cross Entropy Loss, yang sesuai untuk tugas klasifikasi multi-kelas.

Tabel 3: Spesifikasi Detail Arsitektur Model

Parameter	**ViT-Tiny**	**Swin-Tiny**
Model Name	`vit_tiny_patch16_224`	`swin_tiny_patch4_window7_224`
Pre-trained	ImageNet-1k	ImageNet-1k
Patch Size	16×16	4×4
Input Size	$224 \times 224 \times 3$	$224 \times 224 \times 3$
Embed Dim	192	96
Depths	12 (Layers)	[2, 2, 6, 2]
Num Heads	3	[3, 6, 12, 24]
Window Size	-	7
Output Classes	10 (CIFAR-10)	10 (CIFAR-10)

Tabel 4: Konfigurasi Hyperparameter Pelatihan

Hyperparameter	**Nilai / Konfigurasi**
Framework	PyTorch + Timm Library
Input Resolution	224×224 piksel
Optimizer	AdamW
Learning Rate	5×10^{-5}
Batch Size	16
Epochs	10
Loss Function	CrossEntropyLoss
Pre-trained Weights	ImageNet-1k

3.4 Library dan Framework

Eksperimen ini dibangun menggunakan Python versi 3.10. PyTorch adalah struktur utama yang digunakan untuk mengembangkan model Deep Learning.

Tabel 5: Daftar Pustaka dan Framework yang Digunakan

Library/Framework	**Fungsi Utama**
Python	Bahasa pemrograman utama (v3.10)
PyTorch	Framework *Deep Learning* dan komputasi tensor
Torchvision	Akses dataset CIFAR-10 dan transformasi citra
Timm	Penyedia arsitektur model ViT dan Swin Transformer
Scikit-learn	Perhitungan metrik evaluasi (F1-Score, Recall, Precision)
Matplotlib	Visualisasi grafik performa pelatihan
Seaborn	Visualisasi *Confusion Matrix*
Pandas	Manipulasi data dan pembuatan file prediksi CSV
NumPy	Operasi numerik dan manipulasi array

3.5 Spesifikasi Hardware

Tabel 6: Spesifikasi Perangkat Keras (Hardware)

Komponen	Spesifikasi
Platform	Local Machine (Windows)
GPU	NVIDIA GeForce GTX 1650 Max-Q
VRAM	4 GB GDDR6
CPU	Intel Core i5 11400H
RAM	16 GB
CUDA Version	11.6

3.6 Cara Pengukuran Metrik Evaluasi

Untuk mengukur performa model yang telah dilatih menggunakan beberapa metrik evaluasi standar. Metrik ini digunakan untuk memberikan gambaran kuantitatif mengenai seberapa baik model dalam mengklasifikasikan data ke dalam kelas yang benar. Dasar dari perhitungan seluruh metrik ini diambil dari Confusion Matrix.

3.6.1 Confusion Matrix

Confusion Matrix adalah tabel representasi yang membandingkan prediksi model dengan label sebenarnya (*ground truth*). Tabel ini memberikan informasi detail mengenai jumlah prediksi yang benar dan salah untuk setiap kelas. Struktur dasar dari *confusion matrix* untuk klasifikasi biner dapat dilihat pada Tabel 7.

Tabel 7: Representasi Confusion Matrix

Kelas Sebenarnya	**Kelas Prediksi**	
	Positif	**Negatif**
Positif	True Positive (TP)	False Negative (FN)
Negatif	False Positive (FP)	True Negative (TN)

Keterangan komponen dalam tabel tersebut adalah sebagai berikut:

- **True Positive (TP):** Data kelas positif yang diprediksi secara benar sebagai positif.
- **True Negative (TN):** Data kelas negatif yang diprediksi secara benar sebagai negatif.
- **False Positive (FP):** Data kelas negatif yang salah diprediksi sebagai positif (Kesalahan Tipe I).
- **False Negative (FN):** Data kelas positif yang salah diprediksi sebagai negatif (Kesalahan Tipe II).

Berdasarkan nilai-nilai di atas, perhitungan metrik evaluasi dijabarkan sebagai berikut:

3.6.2 Akurasi (*Accuracy*)

Akurasi merupakan rasio jumlah prediksi yang benar (baik positif maupun negatif) terhadap keseluruhan jumlah data. Metrik ini memberikan gambaran umum tentang kinerja model. Rumus akurasi didefinisikan pada Persamaan 2:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

3.6.3 Presisi (*Precision*)

Presisi mengukur tingkat ketepatan model dalam memprediksi kelas positif. Metrik ini menunjukkan seberapa akurat model ketika menyatakan suatu data sebagai positif. Rumus presisi didefinisikan pada Persamaan 3:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

3.6.4 Recall (*Sensitivity*)

Recall atau sensitivitas mengukur kemampuan model untuk menemukan kembali seluruh data positif yang sebenarnya. Metrik ini sangat penting untuk meminimalkan *False Negative*. Rumus *recall* didefinisikan pada Persamaan 4:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

3.6.5 F1-Score

F1-Score adalah rata-rata harmonik (*harmonic mean*) dari Presisi dan *Recall*. Metrik ini memberikan representasi tunggal yang menyeimbangkan kedua metrik tersebut, terutama berguna pada dataset yang tidak seimbang. Rumus *F1-Score* didefinisikan pada Persamaan 5:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

4 Hasil dan Analisis

4.1 Perbandingan Jumlah Parameter

Tabel 8 menyajikan rincian jumlah parameter dan ukuran model antara arsitektur ViT-Tiny dan Swin-Tiny yang digunakan dalam eksperimen ini.

Tabel 8: Perbandingan Jumlah Parameter Model

Metrik	ViT-Tiny	Swin-Tiny
Total Parameters	5,526,346	27,527,044
Trainable Parameters	5,526,346	27,527,044
Non-trainable Parameters	0	0
Model Size (MB)	21.08	105.01

Analisis:

- **Perbedaan Ukuran Signifikan:** Swin-Tiny memiliki jumlah parameter sekitar 5 kali lipat lebih banyak (27.5M) dibandingkan ViT-Tiny (5.5M). Hal ini berdampak langsung pada ukuran file model, di mana Swin-Tiny mencapai 105.01 MB sedangkan ViT-Tiny hanya 21.08 MB.
- **Kompleksitas Arsitektur:** Tingginya jumlah parameter pada Swin Transformer disebabkan oleh struktur hierarkisnya yang kompleks dengan mekanisme *Shifted Windows* dan *Patch Merging* antar layer untuk menangkap fitur multi-skala. Sebaliknya, ViT-Tiny didesain sebagai model yang sangat ringan dengan arsitektur kolumnar sederhana (*isotropic*).
- **Status Parameter:** Seluruh parameter pada kedua model berstatus *trainable* (dapat dilatih), karena pada tahap inisialisasi (sel 4 pada *notebook*), model dimuat lengkap dan optimizer diatur untuk memperbarui seluruh bobot jaringan (`model.parameters()`) selama proses *fine-tuning*.

4.2 Perbandingan Metrik Performa

Setelah proses pelatihan selesai, kedua model dievaluasi menggunakan data uji (*test set*) CIFAR-10 yang belum pernah dilihat sebelumnya. Tabel 9 merangkum hasil evaluasi berdasarkan empat metrik utama: Akurasi, Presisi, *Recall*, dan *F1-Score*.

Tabel 9: Perbandingan Metrik Performa (Weighted Avg)

Metrik	ViT-Tiny	Swin-Tiny
Akurasi (*Accuracy*)	96%	97%
Presisi (*Precision*)	96%	97%
Recall	96%	97%
F1-Score	96%	97%

Analisis:

- **Keunggulan Swin Transformer:** Berdasarkan hasil eksperimen, Swin-Tiny menunjukkan performa yang sedikit lebih unggul dibandingkan ViT-Tiny dengan selisih akurasi sebesar 1%. Hal ini mengindikasikan bahwa mekanisme *Shifted Window Attention* pada Swin lebih efektif dalam menangkap fitur lokal dan global dibandingkan *Global Attention* murni pada ViT untuk resolusi gambar ini.
- **Keseimbangan Presisi dan Recall:** Kedua model menunjukkan nilai Presisi dan *Recall* yang seimbang (selisih sangat kecil atau sama), yang ditunjukkan oleh nilai *F1-Score* yang tinggi. Ini menandakan bahwa model mampu memprediksi kelas dengan tepat tanpa banyak menghasilkan *False Positive* atau *False Negative*.
- **Stabilitas Model:** Tingginya skor di atas 95% pada kedua model menunjukkan bahwa arsitektur *Vision Transformer*, baik varian standar maupun hierarkis, sangat kompeten untuk menangani dataset CIFAR-10, bahkan dengan ukuran model yang tergolong "Tiny".

4.3 Perbandingan Waktu Inferensi

Efisiensi komputasi saat fase pengujian (*inference*) merupakan faktor krusial untuk penerapan model pada perangkat dengan sumber daya terbatas atau sistem *real-time*. Tabel 10 menampilkan perbandingan kecepatan antara ViT-Tiny dan Swin-Tiny dalam memproses 10.000 gambar pada *Test Set* menggunakan perangkat keras yang sama.

Tabel 10: Perbandingan Waktu Inferensi dan Throughput

Metrik	ViT-Tiny	Swin-Tiny
Total Waktu (10.000 gambar)	36.26 detik	117.99 detik
Rata-rata (*Latency*)	3.63 ms/img	11.80 ms/img
Throughput	275.82 img/sec	84.75 img/sec

Analisis:

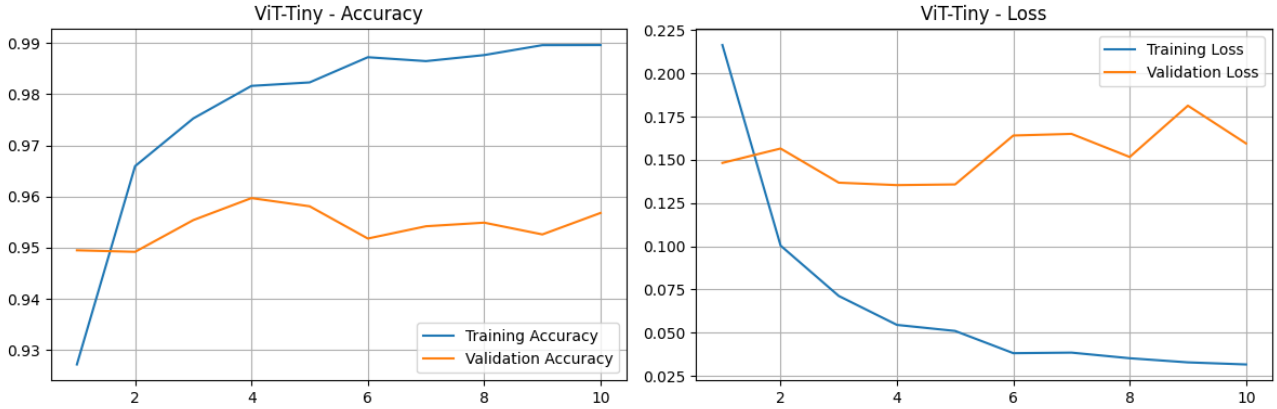
- **Dominasi Kecepatan ViT:** ViT-Tiny menunjukkan performa inferensi yang jauh lebih unggul dengan *throughput* mencapai 275.82 citra per detik. Angka ini sekitar **3.25 kali lebih cepat** dibandingkan Swin-Tiny yang hanya mencapai 84.75 citra per detik.
- **Latency Rendah:** Rata-rata waktu yang dibutuhkan ViT-Tiny untuk memproses satu gambar hanya 3.63 ms, menjadikannya kandidat yang sangat efisien untuk aplikasi yang membutuhkan respon cepat. Sebaliknya, Swin-Tiny membutuhkan waktu lebih lama (11.80 ms) untuk setiap gambar.

- **Trade-off Kompleksitas vs Kecepatan:** Perbedaan kecepatan yang signifikan ini berkorelasi langsung dengan jumlah parameter. Arsitektur Swin yang memiliki parameter 5 kali lebih besar dan mekanisme *shifted window attention* yang lebih rumit membebani proses komputasi, menyebabkan waktu inferensi menjadi lebih lambat meskipun akurasinya sedikit lebih tinggi.

4.4 Visualisasi

4.4.1 Kurva Learning

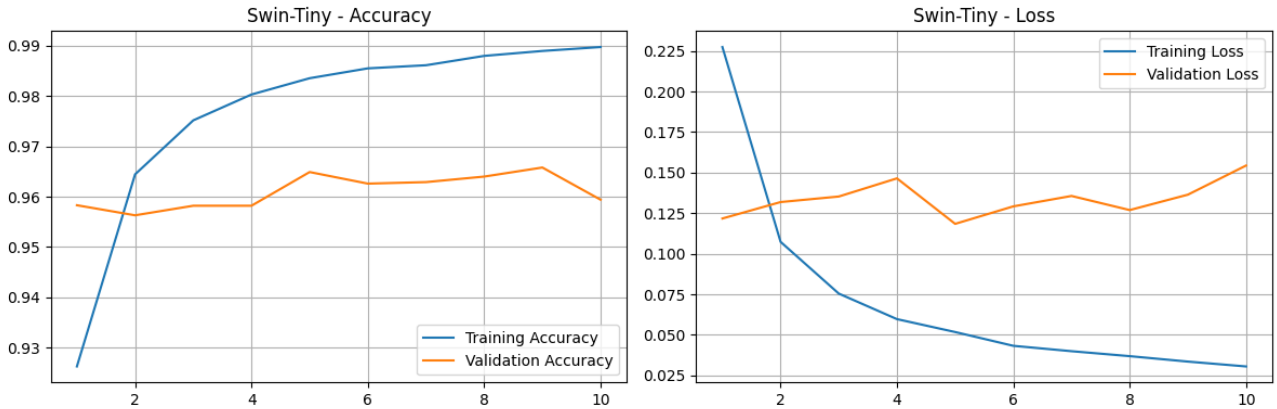
Visualisasi proses pelatihan (*training*) dan validasi untuk kedua model disajikan dalam Gambar 1 dan Gambar 2. Grafik ini menunjukkan pergerakan nilai Akurasi dan *Loss* pada setiap *epoch*.



Gambar 1: Grafik Akurasi dan Loss Model ViT-Tiny selama 10 Epoch

Analisis ViT-Tiny:

- **Training Accuracy:** Meningkat secara stabil dan tajam dari sekitar 92% hingga mencapai hampir 99% pada epoch ke-10, menunjukkan model mampu mempelajari data latih dengan sangat baik.
- **Validation Accuracy:** Mencapai puncaknya di sekitar epoch ke-4 ($\approx$ 96%), namun setelah itu cenderung stagnan dan sedikit berfluktuasi di kisaran 95-96%.
- **Indikasi Overfitting:** Terdapat celah (*gap*) yang mulai melebar antara kurva *Training* dan *Validation* setelah epoch ke-4. Hal ini juga terlihat pada grafik *Loss*, di mana *Training Loss* terus turun, namun *Validation Loss* mulai naik kembali setelah epoch ke-5, mengindikasikan model mulai mengalami *overfitting* ringan terhadap data latih.



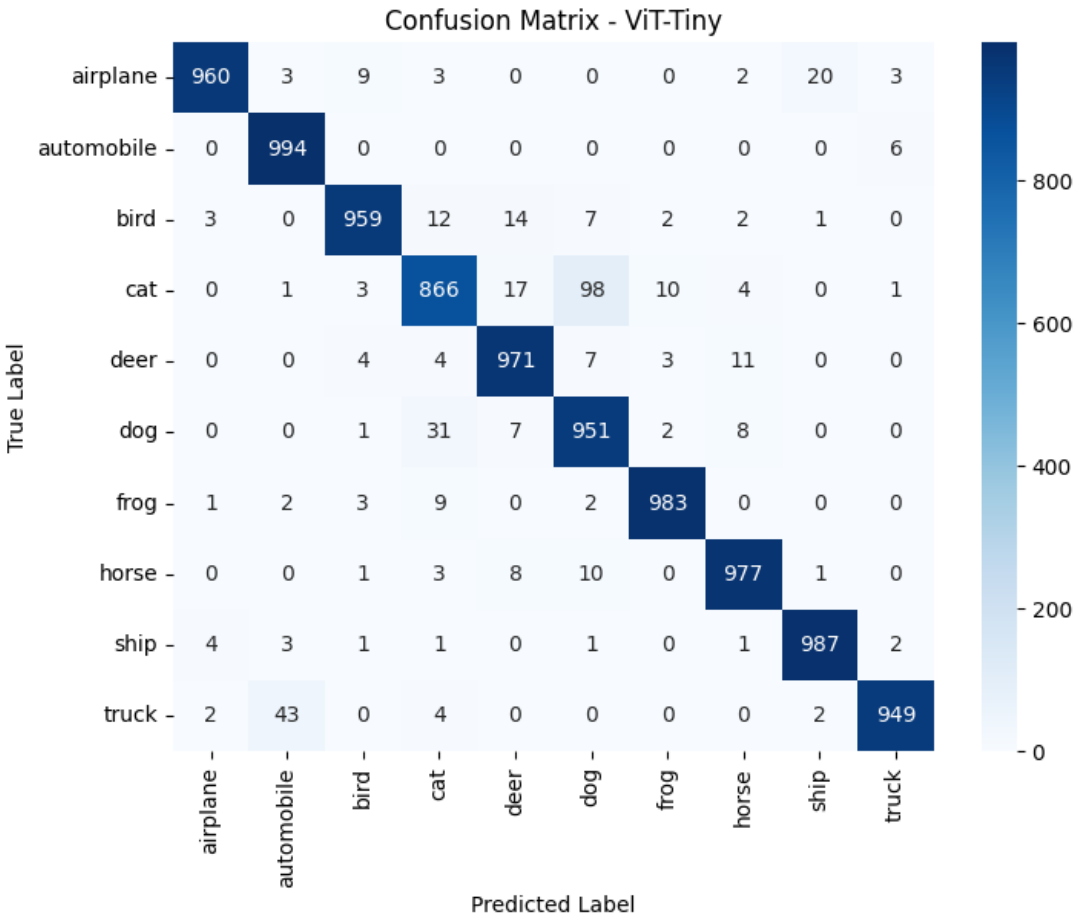
Gambar 2: Grafik Akurasi dan Loss Model Swin-Tiny selama 10 Epoch

Analisis Swin-Tiny:

- **Stabilitas Konvergensi:** Kurva akurasi pada Swin-Tiny menunjukkan pola yang mirip dengan ViT, namun dengan pencapaian *Validation Accuracy* yang sedikit lebih tinggi dan stabil di atas 96% pada pertengahan pelatihan (epoch 5-9).
- **Loss Pattern:** *Training Loss* menurun secara konsisten hingga di bawah 0.05. Sama seperti ViT, grafik *Validation Loss* pada Swin juga mengalami fluktuasi naik setelah epoch ke-5, yang menandakan titik optimal model sebenarnya berada di sekitar epoch 5 atau 6 sebelum *generalization error* meningkat.

4.4.2 Confusion Matrix

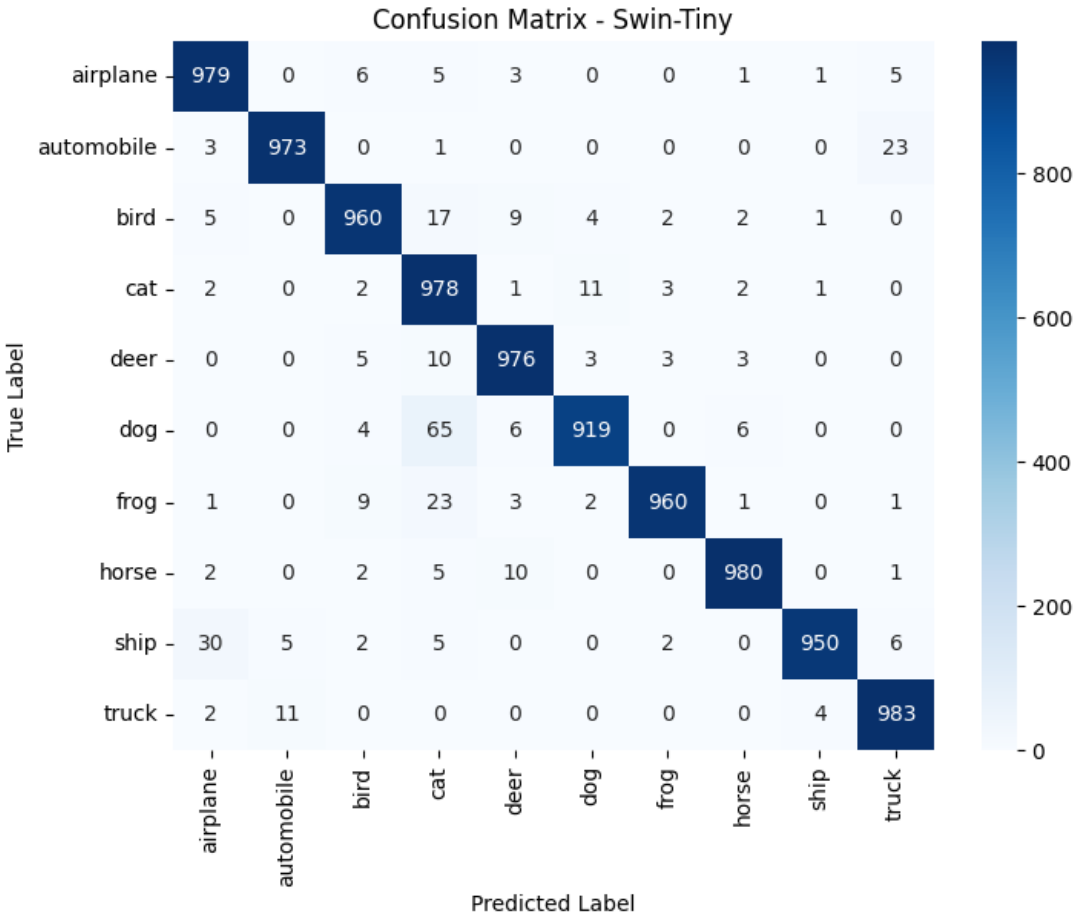
Confusion Matrix digunakan untuk menganalisis performa model secara lebih granular pada setiap kelas. Gambar 3 dan Gambar 4 menampilkan distribusi prediksi untuk 10 kelas pada dataset CIFAR-10.



Gambar 3: Confusion Matrix Model ViT-Tiny

Analisis ViT-Tiny:

- **Kelas Dominan:** ViT-Tiny sangat akurat dalam mengenali kendaraan, terlihat dari angka prediksi benar yang tinggi pada kelas *automobile* (994) dan *ship* (987).
- **Misklafikasi Signifikan:** Kesalahan terbesar terjadi pada kelas *cat* (kucing), di mana 98 gambar salah diprediksi sebagai *dog* (anjing). Hal ini wajar karena kemiripan fitur visual (kaki empat, bulu, bentuk tubuh) antara kedua hewan tersebut. Selain itu, terdapat kebingungan antara *truck* dan *automobile* (43 kesalahan), yang juga memiliki kemiripan struktur visual.



Gambar 4: Confusion Matrix Model Swin-Tiny

Analisis Swin-Tiny:

- **Peningkatan pada Kelas Hewan:** Swin-Tiny menunjukkan perbaikan signifikan dalam membedakan hewan. Kesalahan prediksi *cat* sebagai *dog* menurun drastis menjadi hanya 11 kasus (dibandingkan 98 pada ViT). Ini membuktikan efektivitas mekanisme atensi lokal Swin dalam menangkap detail tekstur yang membedakan spesies hewan.
- **Kesalahan Spesifik:** Meskipun akurasi hewan meningkat, Swin-Tiny memiliki sedikit kelemahan pada kelas *dog*, di mana 65 gambar salah diprediksi sebagai *cat*. Namun, secara keseluruhan, distribusi prediksi pada diagonal utama (prediksi benar) lebih merata dan tebal dibandingkan ViT.

4.5 Analisis Mendalam

4.5.1 Analisis Performa Model

Berdasarkan data dari tabel evaluasi dan pola visual dari kurva pembelajaran, dapat ditarik beberapa analisis mendalam mengenai perilaku kedua arsitektur *Vision Transformer* pada dataset CIFAR-10:

1. **Superioritas Arsitektur Hierarkis (Swin):** Swin-Tiny secara konsisten mengungguli ViT-Tiny dengan akurasi akhir mencapai 97% berbanding 96%. Keunggulan ini dapat dikaitkan dengan mekanisme *Hierarchical Shifted Window Attention* pada Swin Transformer. Mekanisme ini memungkinkan model untuk menangkap fitur lokal (seperti tekstur bulu pada kelas *Cat* dan *Dog*) dengan lebih baik pada layer

awal, sebelum menggabungkannya menjadi informasi semantik global pada layer yang lebih dalam. Sebaliknya, ViT yang menggunakan *Global Self-Attention* sejak awal cenderung kesulitan membedakan fitur-fitur halus pada objek yang memiliki kemiripan visual tinggi.

2. **Efektivitas Transfer Learning:** Kedua model menunjukkan konvergensi yang sangat cepat, mencapai akurasi di atas 90% hanya dalam 2 epoch pertama. Hal ini membuktikan efektivitas penggunaan bobot *pre-trained* ImageNet-1k. Model tidak perlu mempelajari ekstraksi fitur dari nol, melainkan hanya melakukan adaptasi (*fine-tuning*) terhadap domain CIFAR-10. Tanpa *pre-training*, arsitektur berbasis Transformer biasanya membutuhkan ratusan epoch untuk mencapai performa setara karena kurangnya *inductive bias*.

3. **Indikasi Overfitting:** Sebagaimana terlihat pada Gambar Kurva Learning, kedua model mulai menunjukkan gejala *overfitting* setelah epoch ke-5, di mana *Training Loss* terus menurun mendekati nol, namun *Validation Loss* mulai fluktuatif dan sedikit meningkat. Ini menunjukkan bahwa kapasitas model (terutama Swin dengan 28M parameter) sebenarnya terlalu besar untuk dataset CIFAR-10 yang hanya memiliki 50.000 gambar latih, sehingga model mulai "menghafal" data training.

4. **Sensitivitas Terhadap Kelas Sulit:** Analisis *Confusion Matrix* mengungkapkan bahwa ViT-Tiny memiliki kelemahan spesifik dalam membedakan kelas *Cat* dan *Dog*. Kedua kelas ini memiliki variasi pose dan latar belakang yang tinggi. Swin Transformer menangani masalah ini lebih baik karena struktur *window-based attention*-nya membatasi komputasi pada area lokal terlebih dahulu, sehingga lebih robust terhadap gangguan latar belakang (*background noise*) dibandingkan ViT yang langsung melihat keseluruhan gambar secara global.

4.5.2 Trade-off Antara Akurasi, Parameter, dan Kecepatan

Pemilihan arsitektur model untuk implementasi dunia nyata tidak hanya bergantung pada metrik akurasi semata, melainkan keseimbangan (*trade-off*) antara performa prediksi dan biaya komputasi. Berdasarkan data eksperimen, ditemukan hubungan non-linear yang signifikan antara kompleksitas model dan keuntungan performa yang didapatkan:

1. ***The Cost of Accuracy* (Biaya Akurasi):** Swin-Tiny berhasil mengungguli ViT-Tiny dengan selisih akurasi sebesar 1% (97% vs 96%). Namun, peningkatan marginal ini membutuhkan "biaya" sumber daya yang sangat besar. Jumlah parameter meningkat hampir 5 kali lipat (dari 5.5 juta menjadi 27.5 juta) dan ukuran penyimpanan model membengkak dari 21 MB menjadi 105 MB.

2. **Latensi vs Kompleksitas Mekanisme:** Arsitektur *Shifted Window* pada Swin Transformer, meskipun efektif menangkap konteks lokal, menambah beban komputasi yang berat. Waktu inferensi Swin-Tiny (11.80 ms/img) lebih dari 3 kali lebih lambat dibandingkan ViT-Tiny (3.63 ms/img). Hal ini menunjukkan bahwa operasi penggeseran jendela (*window shifting*) dan penggabungan patch (*patch merging*) jauh lebih memakan waktu dibandingkan operasi *MatMul* (Matrix Multiplication) standar pada *Global Self-Attention* milik ViT untuk resolusi gambar kecil (224×224).

3. **Rasio Efisiensi:** Jika dilihat dari rasio *Performance-to-Cost*, ViT-Tiny menawarkan efisiensi yang jauh lebih baik. Model ini mampu memberikan akurasi setara *state-of-the-art* (96%) dengan kecepatan *throughput* yang sangat tinggi (275 img/s). Sebaliknya, Swin-Tiny, meskipun lebih akurat, kurang ideal untuk perangkat dengan daya komputasi rendah (*edge devices*) karena latensinya yang lebih tinggi.

4.5.3 Kesesuaian Model dengan Dataset

Keberhasilan kedua model dalam mencapai akurasi di atas 96% pada dataset CIFAR-10 menunjukkan tingkat kesesuaian yang tinggi, meskipun terdapat perbedaan fundamental antara karakteristik asli arsitektur *Transformer* dengan sifat dataset CIFAR-10. Analisis kesesuaian ini dapat ditinjau dari tiga aspek utama:

1. **Adaptasi Resolusi Input (*Input Resolution Adaptation*):** Dataset CIFAR-10 memiliki resolusi asli 32×32 piksel, yang sangat kecil untuk arsitektur standar ViT yang menggunakan *patch size* 16×16. Jika menggunakan resolusi asli, gambar hanya akan terbagi menjadi 2×2 *patches*, yang tidak cukup untuk menangkap informasi spasial yang bermakna.

   Oleh karena itu, strategi *upscaling* (mengubah ukuran) citra menjadi 224×224 piksel yang diterapkan dalam penelitian ini adalah langkah krusial. Hal ini memungkinkan ViT membagi gambar menjadi 14×14 *patches* (total 196 token), memberikan ruang yang cukup bagi mekanisme *Self-Attention* untuk mempelajari hubungan spasial yang kompleks, serta memungkinkan penggunaan bobot *pre-trained* dari ImageNet tanpa modifikasi struktur.

2. **Peran Vital *Transfer Learning*:** Secara teoritis, *Vision Transformer* memiliki *inductive bias* yang rendah dan membutuhkan dataset skala besar (seperti JFT-300M atau ImageNet-21k) untuk berkinerja baik. CIFAR-10 yang hanya memiliki 50.000 data latih tergolong kecil untuk melatih ViT dari awal (*scratch*).

   Hasil eksperimen membuktikan bahwa teknik *Transfer Learning* (menggunakan bobot *pre-trained* ImageNet-1k) berhasil menjembatani kesenjangan data tersebut. Model tidak perlu mempelajari filter visual dasar (seperti tepi dan tekstur) dari nol, melainkan hanya melakukan *fine-tuning* untuk menyesuaikan representasi fitur yang sudah matang ke 10 kelas CIFAR-10. Tanpa *pre-training*, model kemungkinan besar akan gagal mencapai konvergensi atau mengalami *overfitting* parah.

3. **Kecukupan Kapasitas Model "Tiny":** Pemilihan varian "Tiny" (ViT-Tiny dan Swin-Tiny) terbukti sangat tepat untuk kompleksitas CIFAR-10. Dengan jumlah parameter yang lebih kecil dibandingkan varian "Base" atau "Large", varian Tiny memiliki risiko *overfitting* yang lebih rendah pada dataset berukuran sedang. Swin-Tiny, dengan struktur hierarkisnya, menunjukkan kecocokan alami yang lebih baik untuk menangkap objek dengan variasi skala yang beragam pada CIFAR-10 dibandingkan struktur *columnar* ViT.

5 Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan hasil eksperimen, analisis data, dan evaluasi model yang telah dilakukan dalam membandingkan arsitektur *Vision Transformer* (ViT-Tiny) dan *Swin Transformer* (Swin-Tiny) pada dataset CIFAR-10, dapat ditarik beberapa kesimpulan utama:

1. **Performa Klasifikasi:** Swin Transformer (Swin-Tiny) terbukti memiliki kemampuan generalisasi yang sedikit lebih baik dengan akurasi pengujian sebesar **97%**, mengungguli ViT-Tiny yang mencapai **96%**. Keunggulan ini menegaskan bahwa pendekatan hierarkis dengan mekanisme *Shifted Window Attention* lebih efektif dalam menangkap detail fitur lokal pada citra resolusi rendah hingga menengah dibandingkan pendekatan *Global Self-Attention* murni.

2. **Efisiensi Komputasi dan Kecepatan:** ViT-Tiny menunjukkan superioritas yang signifikan dalam hal efisiensi. Dengan jumlah parameter yang jauh lebih sedikit (5.5 juta vs 27.5 juta), ViT-Tiny mampu memproses data 3.25 kali lebih cepat dengan *throughput* mencapai **275.82 citra/detik**, dibandingkan Swin-Tiny yang hanya mencapai **84.75 citra/detik**.

3. **Analisis Kesalahan (*Error Analysis*):** Berdasarkan *Confusion Matrix*, Swin Transformer lebih robust dalam membedakan kelas yang memiliki kemiripan fitur visual tinggi, seperti antara *Cat* dan *Dog*. Hal ini menunjukkan bahwa fitur lokal yang diekstraksi oleh Swin sangat krusial untuk disambiguasi objek yang memiliki struktur bentuk serupa namun berbeda tekstur.

4. **Trade-off Implementasi:** Pemilihan model terbaik bergantung pada batasan aplikasi. Jika prioritas utama adalah ketepatan prediksi (akurasi), maka **Swin-Tiny** adalah pilihan terbaik. Namun, untuk

implementasi pada perangkat dengan sumber daya terbatas (*edge devices*) atau sistem *real-time* yang membutuhkan latensi rendah, **ViT-Tiny** adalah solusi yang jauh lebih optimal karena menawarkan keseimbangan terbaik antara kecepatan tinggi dan akurasi yang kompetitif.

5.2 Rekomendasi Model

Berdasarkan hasil komparasi kinerja dan efisiensi antara ViT-Tiny dan Swin-Tiny, berikut adalah rekomendasi penggunaan model sesuai dengan skenario kebutuhan aplikasi:

1. **Akurasi Maksimal** Jika sistem yang dikembangkan menuntut tingkat presisi yang maksimal (misalnya untuk analisis medis atau sistem keamanan kritis) di mana kesalahan prediksi harus diminimalisir sekecil mungkin, maka **Swin Transformer (Swin-Tiny)** adalah pilihan yang direkomendasikan. Kemampuannya dalam menangkap detail fitur lokal memberikan sedikit keunggulan dalam ketepatan klasifikasi, meskipun dengan konsekuensi biaya komputasi yang lebih tinggi.

2. **Efisiensi Komputasi** Untuk aplikasi yang akan dijalankan pada perangkat dengan sumber daya terbatas seperti ponsel pintar, kamera CCTV pintar, atau perangkat IoT, **ViT-Tiny** adalah pilihan yang jauh lebih superior. Ukuran modelnya yang kecil (21 MB) sangat hemat memori penyimpanan, dan arsitekturnya yang ringan tidak akan membebani prosesor perangkat.

3. **Aplikasi Real-time** Jika sistem membutuhkan respons instan atau harus memproses aliran video dengan *frame rate* tinggi, **ViT-Tiny** sangat direkomendasikan. Dengan *throughput* mencapai 275 gambar per detik, model ini mampu menangani inferensi video secara *real- time* tanpa hambatan (*lag*), jauh melampaui Swin-Tiny yang hanya mencapai 84 gambar per detik.

5.3 Saran untuk Pengembangan Lebih Lanjut

Eksperimen ini telah memberikan wawasan mendasar mengenai perbandingan performa ViT dan Swin Transformer pada citra resolusi rendah. Untuk meningkatkan validitas dan performa di masa mendatang, disarankan beberapa langkah pengembangan berikut:

1. **Penerapan Teknik Regularisasi Lanjutan:** Mengingat adanya indikasi *overfitting* pada kurva pembelajaran setelah *epoch* ke-5, penelitian selanjutnya disarankan untuk menerapkan teknik regularisasi yang lebih agresif. Penggunaan metode augmentasi data seperti *MixUp*, *CutMix*, atau *RandAugment* dapat membantu model menjadi lebih robust dan mengurangi kesenjangan antara *training loss* dan *validation loss*.

2. **Eksplorasi Hyperparameter Tuning:** Eksperimen ini menggunakan *learning rate* statis. Penggunaan *Learning Rate Scheduler* (seperti *Cosine Annealing* atau *Warm-up steps*) sangat disarankan untuk membantu optimizer mencapai titik konvergensi yang lebih optimal dan stabil, berpotensi meningkatkan akurasi akhir di atas 97%.

3. **Pengujian pada Dataset Resolusi Tinggi:** Keunggulan arsitektur Swin Transformer terletak pada efisiensi linear terhadap resolusi gambar. Oleh karena itu, disarankan untuk menguji kedua model pada dataset dengan resolusi asli yang lebih tinggi (seperti dataset medis atau citra satelit) untuk melihat apakah kesenjangan performa dan efisiensi antara ViT dan Swin menjadi lebih signifikan dibandingkan pada dataset CIFAR-10.

4. **Komparasi Varian Model (Scaling Up):** Penelitian ini terbatas pada varian "Tiny". Eksperimen lanjutan dapat membandingkan varian "Small" atau "Base" untuk menganalisis apakah *trade-off* antara akurasi dan kecepatan tetap konsisten saat kapasitas model diperbesar.

References

[1] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 11 976–11 986.

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021. [Online]. Available: https://arxiv.org/abs/2010.11929

[3] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?" in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 12 116–12 128. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/652cf38361a209088302ba2b8b7f51e0-Paper.pdf

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[5] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 10 012–10 022.

A Lampiran

A.1 Source Code

Source code lengkap untuk eksperimen ini dapat diakses pada Github repository:
https://github.com/ArkanHariz/VisionTransformer-Comparison

A.2 Output Training Log

Training Log ViT-Tiny

```
Epoch 1/10
--------------------
   > Batch 500/3125 processed...
   > Batch 500/3125 processed...
   > Batch 1000/3125 processed...
   > Batch 1000/3125 processed...
   > Batch 1500/3125 processed...
   > Batch 1500/3125 processed...
   > Batch 2000/3125 processed...
   > Batch 2000/3125 processed...
   > Batch 2500/3125 processed...
   > Batch 2500/3125 processed...
   > Batch 3000/3125 processed...
   > Batch 3000/3125 processed...
   Train Loss: 0.2164 Acc: 0.9272
   Val Loss:   0.1482 Acc: 0.9495
   Model terbaik disimpan (Acc: 0.9495)
```

Training Log Swin-Tiny

```
    Epoch 1/10
--------------------
   > Batch 500/3125 processed...
   > Batch 500/3125 processed...
   > Batch 1000/3125 processed...
   > Batch 1000/3125 processed...
   > Batch 1500/3125 processed...
   > Batch 1500/3125 processed...
   > Batch 2000/3125 processed...
   > Batch 2000/3125 processed...
   > Batch 2500/3125 processed...
   > Batch 2500/3125 processed...
   > Batch 3000/3125 processed...
   > Batch 3000/3125 processed...
   Train Loss: 0.2274 Acc: 0.9262
   Val Loss:   0.1218 Acc: 0.9583
   Model terbaik disimpan (Acc: 0.9583)
```

A.3 Link Bantuan AI

- https://gemini.google.com/share/7bb72b176eec

- https://gemini.google.com/share/d7aa533aacdd
- https://gemini.google.com/share/7b2b9a45bd6a