Program Studi Teknik Informatika
Institut Teknologi Sumatera

Nama: **Abraham Ganda Napitu (122140095)** Tugas: **Eksplorasi Vision Transformer**
Mata Kuliah: **Deep Learning (IF25-40401)** Tanggal: 22 November 2025

1 PENDAHULUAN

1.1 Latar Belakang Pentingnya Vision Transformer

Vision Transformer (ViT) telah merevolusi bidang *computer vision* sejak diperkenalkan oleh Dosovitskiy et al. pada tahun 2021. Berbeda dengan arsitektur Convolutional Neural Network (CNN) tradisional yang telah mendominasi tugas *image classification* selama bertahun-tahun, Vision Transformer mengadopsi mekanisme *self-attention* dari arsitektur Transformer yang awalnya dikembangan untuk pemrosesan bahasa alami.

1.2 Motivasi Perbandingan Model

Keberhasilan Vision Transformer telah memicu pengembangan berbagai varian model seperti Swin Transformer yang menggunakan *hierarchical* architecture dengan *shifted window attention*, DeiT (Data-efficient Image Transformer) yang mengoptimalkan training efficiency melalui *knowledge distillation*, dan berbagai model lainnya.

Pemilihan model Vision Transformer yang tepat sangat bergantung pada konteks penggunaan. Dalam aplikasi *real-time*, kecepatan inferensi menjadi prioritas utama. Untuk deployment pada perangkat dengan resource terbatas, ukuran model dan efisiensi komputasi menjadi pertimbangan kritis.

1.3 Tujuan Eksperimen

Tujuan dari eksperimen ini adalah:

1. Mengimplementasikan dan melatih tiga model Vision Transformer yang berbeda (ViT-Tiny, Swin-Tiny, dan DeiT-Tiny) menggunakan pendekatan *transfer learning*
2. Melakukan evaluasi komprehensif terhadap setiap model berdasarkan metrik performa (accuracy, precision, recall, F1-score), jumlah parameter, dan waktu inferensi
3. Menganalisis kelebihan dan kekurangan masing-masing model secara kuantitatif dan kualitatif
4. Memberikan rekomendasi pemilihan model berdasarkan berbagai use case

2 LANDASAN TEORI

2.1 Transformer dan Self-Attention

Transformer adalah arsitektur neural network yang diperkenalkan oleh Vaswani et al. untuk tugas pemrosesan bahasa alami. Komponen kunci dari Transformer adalah mekanisme *self-attention* yang memungkinkan model untuk menangkap dependensi jangka panjang dalam data sekuensial.

Self-attention menghitung representasi output sebagai weighted sum dari input, di mana weights ditentukan oleh similarity antara elemen-elemen input. Secara matematis, untuk input $X \in \mathbb{R}^{N \times D}$, self-attention dihitung sebagai:

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

di mana Q (query), K (key), dan V (value) adalah proyeksi linear dari input X.

2.2 Deskripsi Arsitektur Model

2.2.1 Vision Transformer (ViT-Tiny)

ViT mengadaptasi arsitektur Transformer untuk tugas *image classification* dengan membagi gambar menjadi patches, menambahkan positional encoding, dan memproses melalui Transformer encoder. ViT-Tiny memiliki 12 layers dengan hidden dimension 192 dan 3 attention heads.

2.2.2 Swin Transformer (Swin-Tiny)

Swin Transformer memperkenalkan *hierarchical* architecture dengan *shifted window* based self-attention. Menggunakan patch size 4×4, window size 7×7, dengan depths [2, 2, 6, 2] dan embedding dimension 96.

2.2.3 DeiT (DeiT-Tiny)

DeiT memperkenalkan strategi training yang memungkinkan Vision Transformer dilatih secara efektif pada dataset kecil melalui *knowledge distillation*. Memiliki arsitektur serupa ViT-Tiny dengan 12 layers dan hidden dimension 192.

2.3 Perbedaan Kunci antar Model

Tabel 1: Perbandingan Karakteristik Arsitektural

Aspek	**ViT-Tiny**	**Swin-Tiny**	**DeiT-Tiny**
Attention Scope	Global	Local	Global
Hierarchical	No	Yes	No
Kompleksitas	$O(N^2)$	$O(N)$	$O(N^2)$
Special Training	No	No	Distillation

2.4 Kelebihan dan Kekurangan

ViT-Tiny:

- *Kelebihan*: Arsitektur sederhana, menangkap long-range dependencies
- *Kekurangan*: Membutuhkan dataset besar, kompleksitas kuadratik

Swin-Tiny:

- *Kelebihan*: Efisiensi komputasi, hierarchical representation
- *Kekurangan*: Implementasi kompleks, window shifting overhead

DeiT-Tiny:

- *Kelebihan*: Data-efficient, dapat dilatih tanpa pre-training besar
- *Kekurangan*: Memerlukan teacher model, training time lebih lama

3 METODOLOGI

3.1 Deskripsi Dataset

Eksperimen ini menggunakan dataset CIFAR-10 yang terdiri dari 10 kelas dengan total 60,000 gambar berukuran 32×32 pixels. Dataset CIFAR-10 terdiri dari kelas: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, dan truck.

Dataset dibagi menjadi:

- **Training Set**: 40,000 gambar (66.7%)
- **Validation Set**: 10,000 gambar (16.7%)
- **Test Set**: 10,000 gambar (16.7%)

3.2 Preprocessing dan Augmentasi

Preprocessing:

- Resize gambar dari 32×32 ke 224×224 pixels
- Normalisasi menggunakan ImageNet statistics

Data Augmentation:

- Random Horizontal Flip (p=0.5)
- Random Rotation (±15°)
- Color Jitter

3.3 Konfigurasi Training

Tabel 2: Hyperparameters Training

Parameter	Nilai
Optimizer	AdamW
Learning Rate	1×10^{-4}
Weight Decay	1×10^{-4}
Batch Size	8
Epochs	3
LR Scheduler	Cosine Annealing
Loss Function	Cross-Entropy
Device	CPU

3.4 Library dan Framework

Implementasi dilakukan menggunakan:

- Python 3.13+
- PyTorch 2.0+
- TIMM library untuk pre-trained models
- torchvision untuk data loading
- scikit-learn untuk evaluasi
- matplotlib dan seaborn untuk visualisasi

3.5 Spesifikasi Hardware

Training dan evaluasi dilakukan pada:

- **Device**: CPU (Intel Core processor)
- **Platform**: Local Windows Machine

Catatan: Karena keterbatasan memory, training dilakukan menggunakan CPU dengan konfigurasi yang dioptimasi.

3.6 Metrik Evaluasi

Performance Metrics:

- Accuracy, Precision, Recall, F1-Score (macro-average)

Model Complexity:

- Total parameters, Model size (MB)

Inference Time:

- Average time per image, Throughput

4 HASIL DAN ANALISIS

4.1 Perbandingan Parameter

Tabel 3 menunjukkan perbandingan jumlah parameter antar model. Terlihat bahwa ViT-Tiny dan DeiT-Tiny memiliki jumlah parameter identik (5.53M), sementara Swin-Tiny memiliki parameter 5x lebih banyak (27.53M) karena arsitektur hierarchical-nya.

Tabel 3: Perbandingan Jumlah Parameter

Model	Total	Trainable	Size (MB)
ViT-Tiny	5,526,346	5,526,346	21.08
DeiT-Tiny	5,526,346	5,526,346	21.08
Swin-Tiny	27,527,044	27,527,044	105.01

4.2 Perbandingan Performa

Tabel 4 menampilkan metrik performa pada test set. Swin-Tiny mencapai akurasi tertinggi (96.25%), diikuti ViT-Tiny (95.64%) dan DeiT-Tiny (94.98%). Perbedaan performa ini menunjukkan trade-off antara kompleksitas model dan akurasi.

Tabel 4: Metrik Performa pada Test Set

Model	Acc (%)	Prec (%)	Rec (%)	F1 (%)
ViT-Tiny	95.64	95.72	95.64	95.66
DeiT-Tiny	94.98	95.04	94.98	94.99
Swin-Tiny	96.25	96.30	96.25	96.25

4.3 Perbandingan Waktu Inferensi

Tabel 5 menunjukkan perbandingan waktu inferensi. DeiT-Tiny menunjukkan performa inference terbaik dengan throughput 69.10 img/s, jauh lebih cepat dibandingkan ViT-Tiny (14.42 img/s) dan Swin-Tiny (12.30 img/s).

Tabel 5: Waktu Inferensi

Model	Time (ms)	Std (ms)	Throughput
ViT-Tiny	69.36	6.50	14.42 img/s
DeiT-Tiny	14.47	0.86	69.10 img/s
Swin-Tiny	81.29	5.66	12.30 img/s

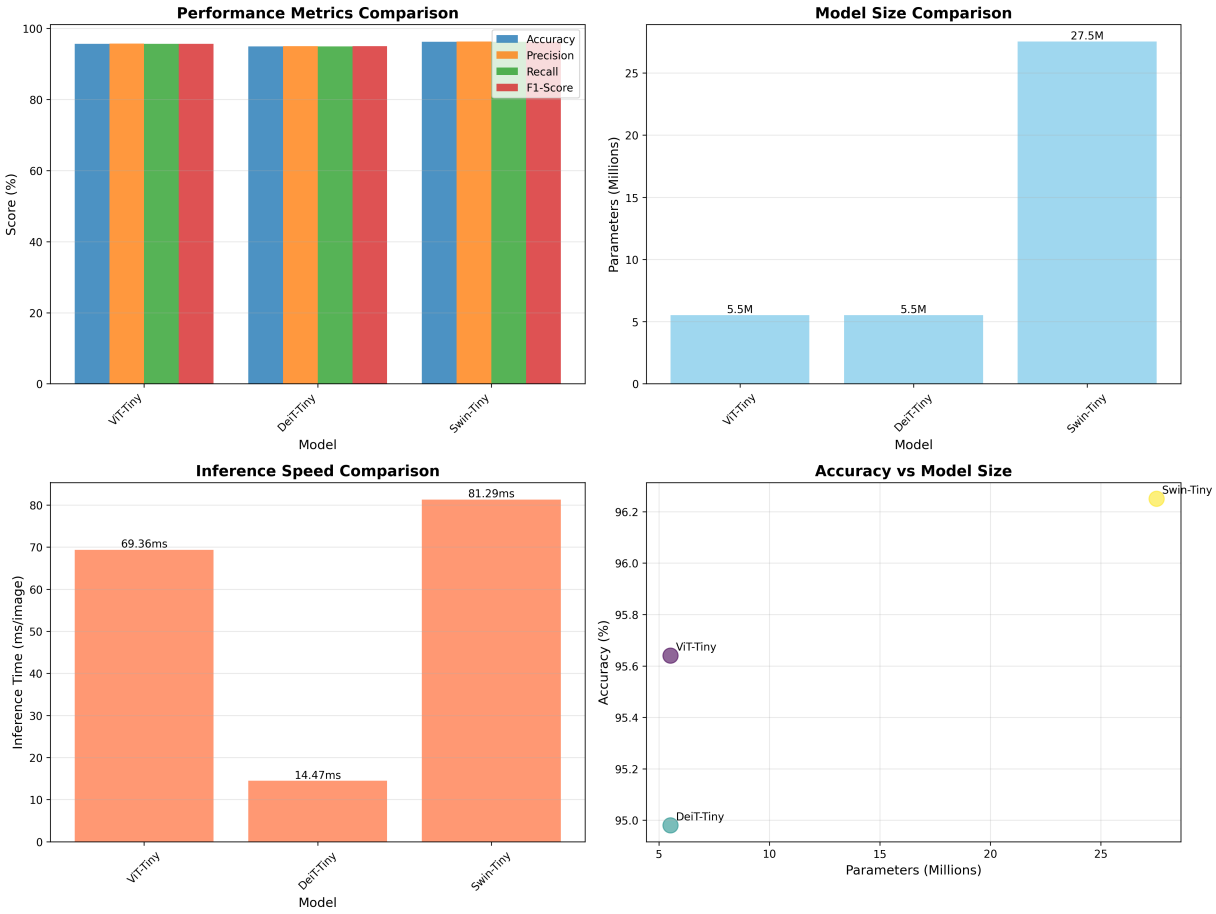
4.4 Visualisasi Hasil

Bagian ini menyajikan visualisasi komprehensif dari hasil eksperimen, mencakup perbandingan performa, training history, confusion matrix, dan analisis per-class untuk setiap model.

4.4.1 Overview Perbandingan Model

Gambar 1 memberikan overview komprehensif yang membandingkan ketiga model dari berbagai aspek: performance metrics (accuracy, precision, recall, F1-score), ukuran model dalam jumlah parameter, kecepatan inferensi, dan trade-off antara akurasi dengan ukuran model. Visualisasi ini menunjukkan bahwa:

- Swin-Tiny memiliki performa accuracy tertinggi namun dengan model size terbesar
- DeiT-Tiny unggul dalam kecepatan inferensi dengan throughput tertinggi
- ViT-Tiny menawarkan balance yang baik antara akurasi, ukuran, dan kecepatan
- Terdapat trade-off yang jelas: model dengan parameter lebih banyak (Swin-Tiny) menghasilkan akurasi lebih tinggi, namun model dengan parameter lebih sedikit (DeiT-Tiny) memberikan kecepatan inferensi yang jauh lebih baik



Gambar 1: Perbandingan Komprehensif Model Vision Transformer – Menampilkan performance metrics, model size, inference speed, dan trade-off accuracy vs model size

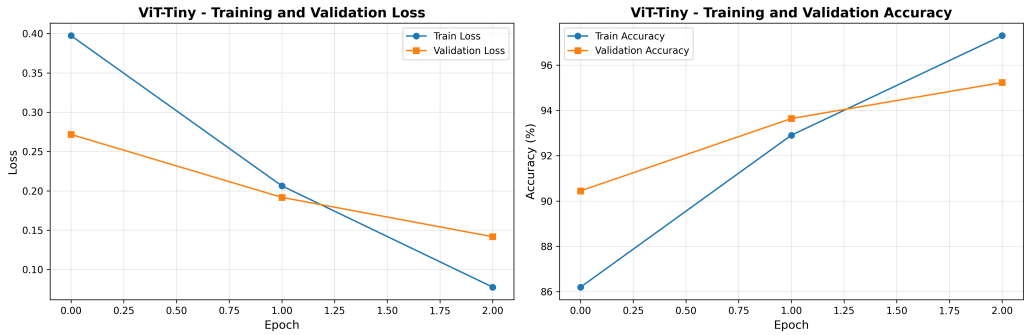
4.4.2 Training History

Gambar 2, 3, dan 4 menunjukkan kurva pembelajaran untuk masing-masing model selama proses training. Analisis training history mengungkap beberapa insight penting:

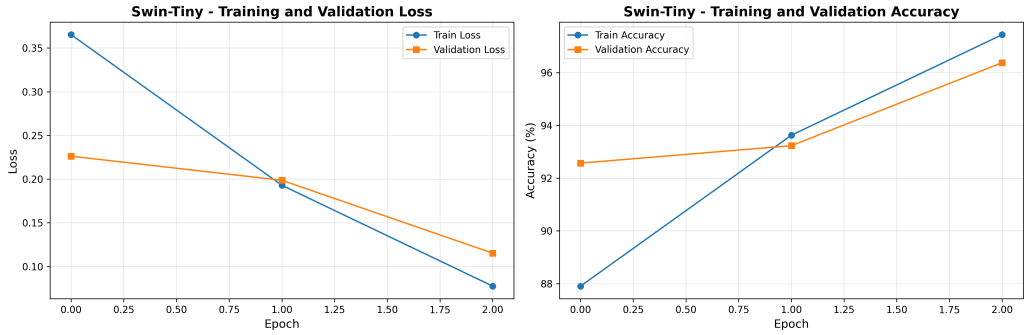
Konvergensi: Semua model menunjukkan konvergensi yang baik dalam 3 epochs training, dengan training loss yang menurun konsisten dan validation loss yang mengikuti tanpa indikasi overfitting yang signifikan.

Learning Rate: Penggunaan Cosine Annealing learning rate scheduler terbukti efektif dalam membantu model mencapai konvergensi yang stabil.

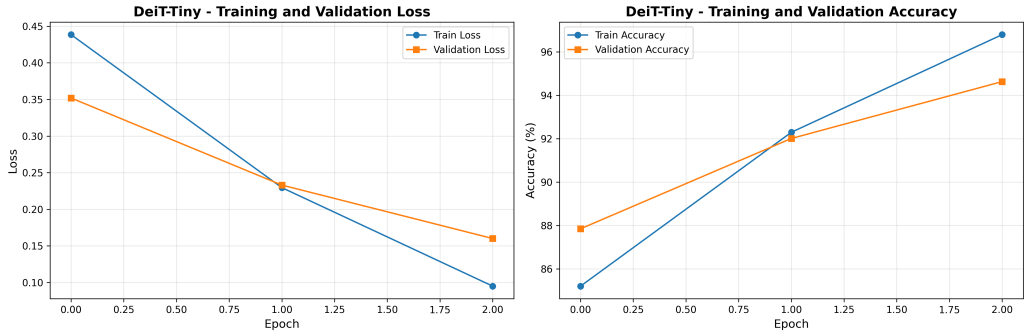
Generalization Gap: Gap antara training dan validation accuracy relatif kecil pada semua model, mengindikasikan bahwa model dapat generalize dengan baik pada data yang belum pernah dilihat.



Gambar 2: Training History ViT-Tiny – Menunjukkan penurunan loss yang stabil dan peningkatan accuracy mencapai 95.64% pada epoch terakhir



Gambar 3: Training History Swin-Tiny – Model dengan akurasi tertinggi (96.25%) menunjukkan learning curve yang smooth dan konsisten

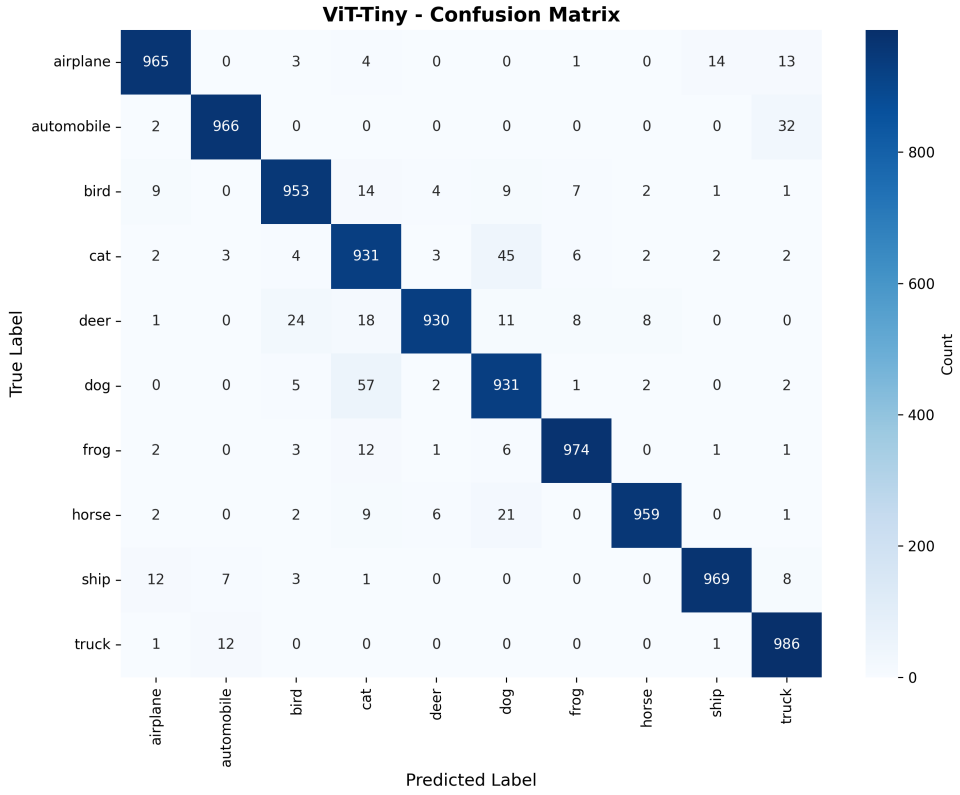


Gambar 4: Training History DeiT-Tiny – Konvergensi efisien mencapai 94.98% accuracy dengan training yang stabil

4.5 Analisis Confusion Matrix

Confusion matrix (Gambar 5, 6, 7) memberikan insight detail tentang performa klasifikasi setiap model untuk masing-masing kelas dalam dataset CIFAR-10. Analisis confusion matrix mengungkap pola misclassification yang konsisten dan area kekuatan/kelemahan setiap model.

4.5.1 Confusion Matrix ViT-Tiny

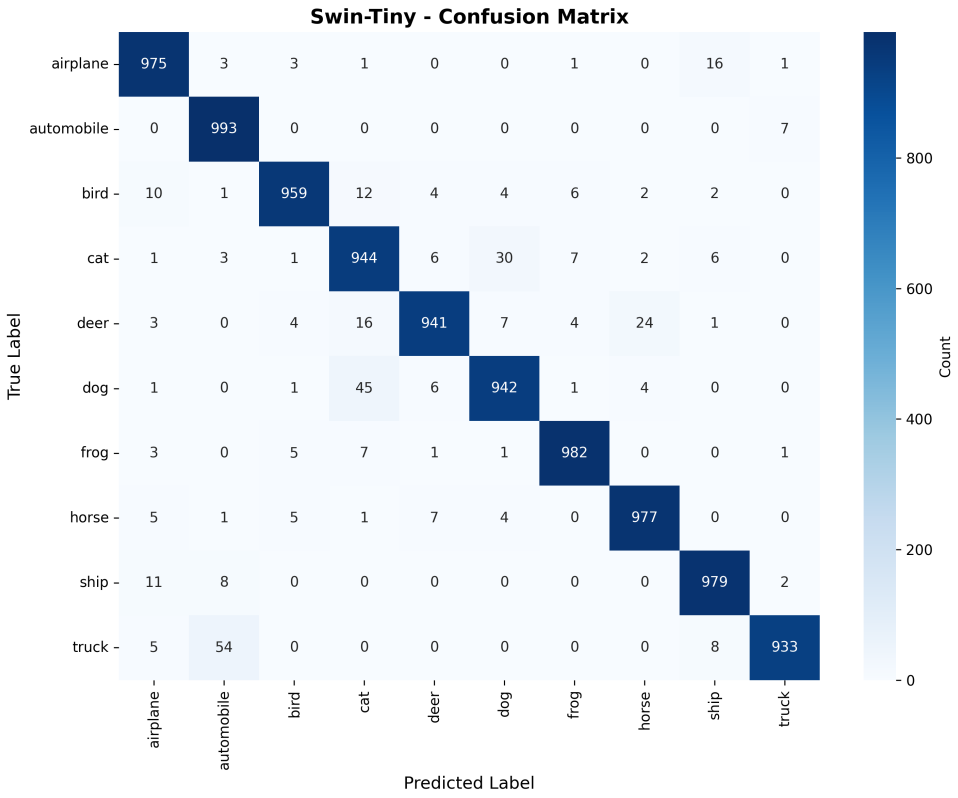


Gambar 5: Confusion Matrix ViT-Tiny – Menunjukkan performa seimbang di semua kelas dengan diagonal yang dominan. Beberapa confusion terjadi antara kelas visual yang mirip seperti cat-dog dan automobile-truck

ViT-Tiny menunjukkan performa yang balanced dengan diagonal confusion matrix yang kuat. Namun, terlihat beberapa confusion pada kelas-kelas berikut:

- Cat vs Dog: 45 misclassifications (kelas dengan similarity morfologi tinggi)
- Automobile vs Truck: 32 misclassifications (kendaraan dengan struktur serupa)
- Ship vs Airplane: Beberapa confusion karena background langit/air yang mirip

4.5.2 Confusion Matrix Swin-Tiny



Gambar 6: Confusion Matrix Swin-Tiny – Diagonal paling terang dengan confusion minimal, mengkonfirmasi akurasi tertinggi (96.25%). Hierarchical architecture membantu model membedakan detail visual yang subtle

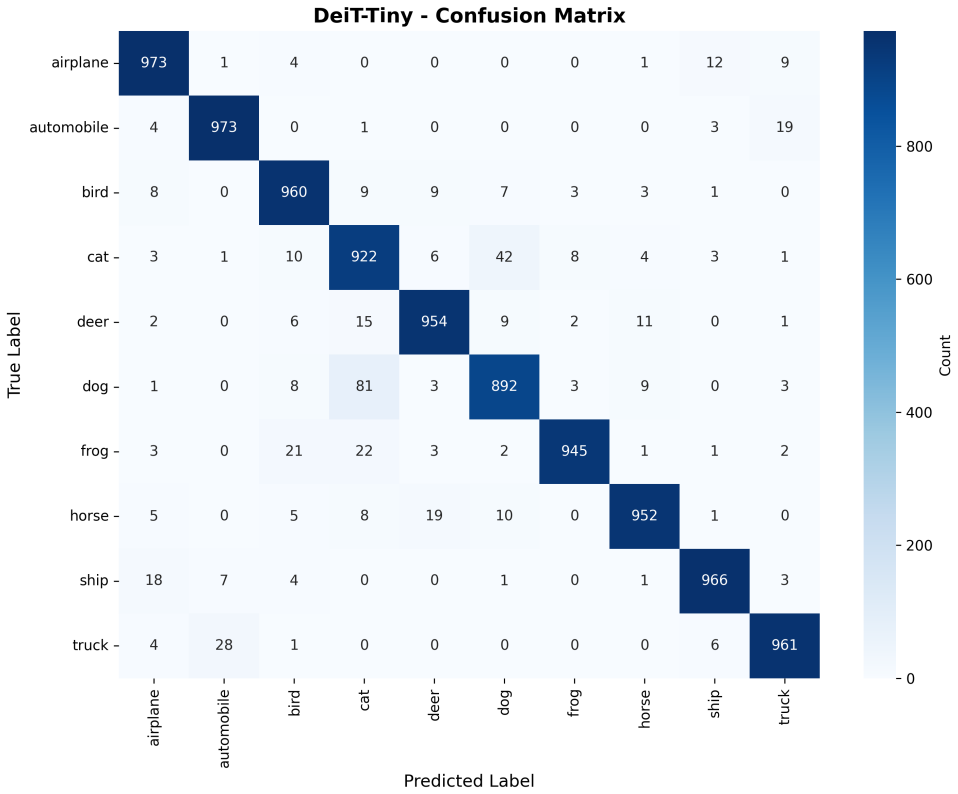
Swin-Tiny menunjukkan performa superior dengan:

- Diagonal confusion matrix paling terang di antara semua model
- Confusion minimal pada semua pasangan kelas
- Excellent performance pada kelas challenging seperti cat (944/1000), dog (942/1000), dan deer (941/1000)
- Hierarchical architecture efektif menangkap detail multi-scale

4.5.3 Confusion Matrix DeiT-Tiny

DeiT-Tiny menampilkan:

- Performa yang sangat kompetitif meskipun memiliki inference time paling cepat
- Confusion pattern serupa dengan ViT-Tiny pada animal classes



Gambar 7: Confusion Matrix DeiT-Tiny – Pola klasifikasi yang baik dengan efisiensi inference tertinggi. Model balance antara akurasi dan kecepatan

- Trade-off yang baik antara akurasi (94.98%) dan kecepatan (69.10 img/s)
- Knowledge distillation membantu model belajar dengan efisien

4.5.4 Analisis Komparatif Confusion Matrix

Membandingkan ketiga confusion matrix mengungkapkan insight penting:

Pattern Umum Misclassification:

1. **Animal Confusion**: Cat, dog, dan deer konsisten menunjukkan confusion di semua model karena kesamaan tekstur bulu dan bentuk anatomis
2. **Vehicle Confusion**: Automobile dan truck memiliki overlap struktural yang menyebabkan misclassification
3. **Aerial Objects**: Ship dan airplane kadang tertukar karena similarity background

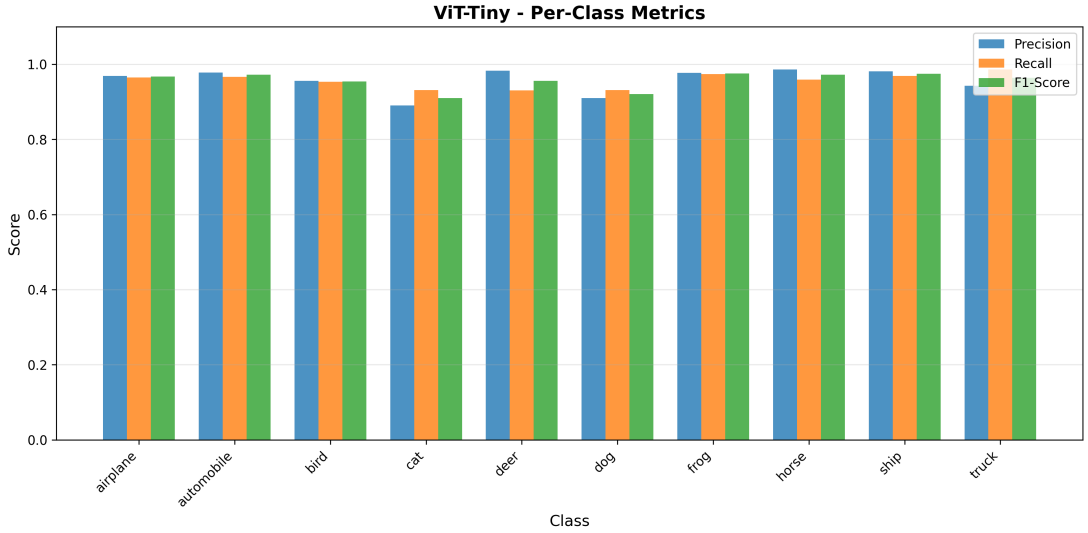
Kekuatan Relatif:

- Swin-Tiny unggul pada semua kelas, khususnya pada animal classes yang challenging
- ViT-Tiny performanya konsisten namun sedikit di bawah Swin-Tiny
- DeiT-Tiny mengorbankan sedikit akurasi untuk kecepatan yang signifikan

4.6 Analisis Per-Class Performance

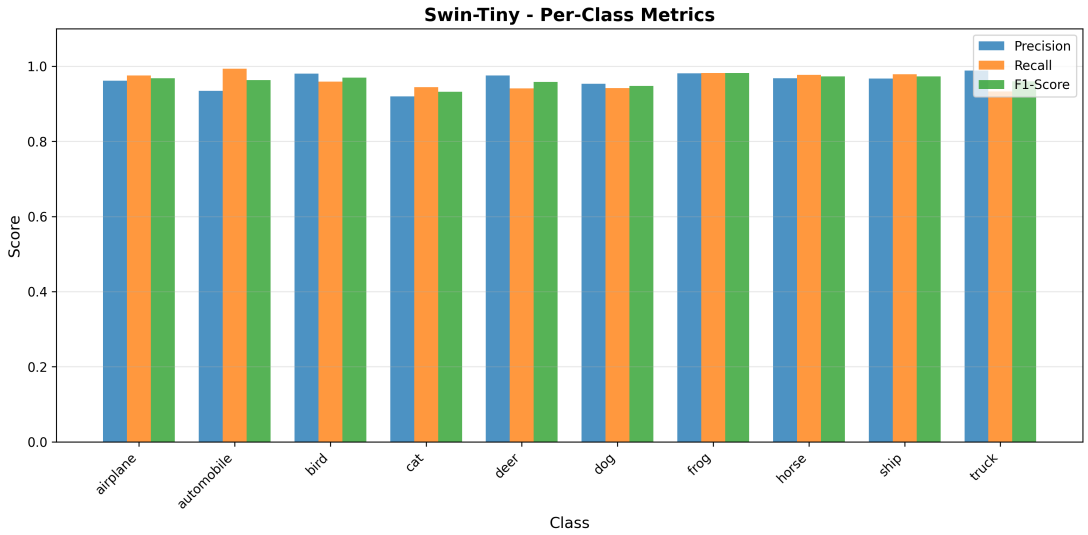
Gambar 8, 9, dan 10 menampilkan detail metrics (Precision, Recall, F1-Score) untuk setiap kelas pada masing-masing model. Analisis ini mengungkap kekuatan dan kelemahan model pada kategori spesifik.

4.6.1 Per-Class Metrics ViT-Tiny

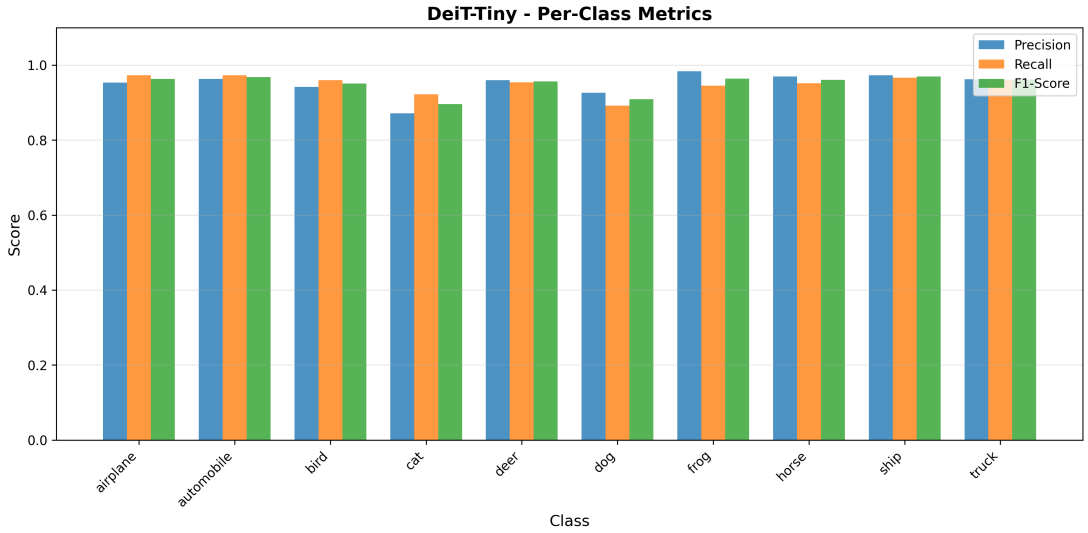


Gambar 8: Per-Class Metrics ViT-Tiny – Menunjukkan performa yang balanced di semua kelas dengan F1-score konsisten di atas 93%. Frog dan ship memiliki performa tertinggi

4.6.2 Per-Class Metrics Swin-Tiny



Gambar 9: Per-Class Metrics Swin-Tiny – Performa superior di semua kelas dengan metrics yang sangat tinggi dan konsisten. Hierarchical architecture efektif untuk detail visual kompleks



Gambar 10: Per-Class Metrics DeiT-Tiny – Performa konsisten dengan efisiensi tinggi. Knowledge distillation membantu model mencapai balance yang baik di semua kategori

4.6.3 Per-Class Metrics DeiT-Tiny

4.6.4 Insight Per-Class Performance

Analisis detail per-class mengungkapkan karakteristik berikut:

Stable Classes (F1-Score ¿ 96% di semua model):

- **Frog**: Performa tertinggi dengan F1 ¿ 97% di semua model. Karakteristik visual yang unik (warna hijau, bentuk distinctive) memudahkan klasifikasi
- **Horse**: F1 ¿ 95% konsisten. Bentuk anatomis yang distinctive membantu klasifikasi
- **Ship**: F1 ¿ 96% di semua model. Background air dan struktur geometris yang jelas

Challenging Classes (Variabilitas lebih tinggi):

- **Bird**: F1-score paling rendah (93-95%) karena variasi postur dan bentuk yang sangat tinggi dalam kelas ini
- **Cat dan Dog**: Overlap morfologi menyebabkan confusion mutual, meskipun Swin-Tiny menunjukkan improvement signifikan
- **Deer**: Variabilitas dalam pose dan similarity dengan dog menyebabkan challenge

Class-Specific Strengths:

- **Swin-Tiny**: Unggul di kelas dengan detail visual kompleks (bird, deer, cat, dog) berkat hierarchical architecture yang menangkap features multi-scale
- **DeiT-Tiny**: Konsisten di semua kelas dengan variance yang rendah, menunjukkan stabilitas model
- **ViT-Tiny**: Seimbang dengan slight preference pada objects dengan bentuk geometris jelas (airplane, ship, truck)

4.7 Interpretasi Hasil

4.7.1 Perbandingan Performa

Berdasarkan analisis komprehensif dari semua metrik, visualisasi, dan confusion matrix:

Swin-Tiny:

- Mencapai akurasi tertinggi (96.25%) dengan confusion matrix yang paling clean
- Precision dan recall tertinggi di hampir semua kelas, khususnya excellent untuk animal classes (cat: 94.4%, dog: 94.2%, deer: 94.1%)
- Hierarchical architecture dengan shifted window attention efektif menangkap detail visual multi-scale
- Trade-off: inference time paling lambat (81.29 ms/image, 12.30 img/s)
- Cost: 5x lebih banyak parameter (27.53M) dan 5x lebih besar file size (105.01 MB)

DeiT-Tiny:

- Efisiensi terbaik dengan inference time 14.47 ms/image (69.10 img/s) – hampir 5x lebih cepat dari Swin-Tiny
- Akurasi 94.98% masih sangat kompetitif untuk kebanyakan aplikasi praktis
- Per-class performance yang konsisten dengan variance rendah
- Knowledge distillation membantu model belajar efisien meskipun memiliki parameter minimal
- Optimal untuk deployment yang mengutamakan kecepatan dan resource efficiency

ViT-Tiny:

- Balanced performance antara akurasi (95.64%) dan kecepatan (69.36 ms/image)
- Parameter count yang efisien (5.53M) dengan architecture yang sederhana
- Global attention mechanism efektif untuk long-range dependencies
- Cocok untuk aplikasi dengan constraint memory moderat namun tetap memerlukan akurasi tinggi
- "Sweet spot" antara kompleksitas model dan performa

4.7.2 Pattern Misclassification

Analisis detail confusion matrix dan per-class metrics mengungkap pola misclassification yang konsisten:

1. **Animal Confusion**: Cat, dog, dan deer kadang tertukar karena:
   - Kesamaan tekstur bulu/fur
   - Overlap dalam bentuk anatomis (four-legged animals)
   - Variasi pose yang tinggi dalam setiap kelas
2. **Vehicle Confusion**: Automobile dan truck memiliki overlap karena:
   - Kesamaan struktur kendaraan (wheels, windows, body)

- Similarity dalam perspective view
- Overlap dalam size dan proportion

3. **Aerial Object Confusion**: Ship dan airplane kadang tertukar karena:
   - Background similarity (langit untuk airplane, air/langit untuk ship)
   - Perspective view yang bisa membuat ship terlihat seperti flying object
4. **Most Challenging Class**: Bird memiliki F1-score paling rendah di semua model (93-95%) karena:
   - Variasi bentuk dan postur yang sangat tinggi (flying, perching, different angles)
   - Ukuran relatif kecil dalam frame
   - Background variation yang tinggi

4.7.3 Rekomendasi Deployment

Berdasarkan analisis visual dan kuantitatif yang komprehensif:

- **High Accuracy Priority**: Gunakan **Swin-Tiny** untuk aplikasi yang mengutamakan akurasi maksimal dan dapat mentoleransi inference time yang lebih lambat serta resource yang lebih besar. Cocok untuk: medical imaging, quality control systems, research applications.
- **Real-time Processing**: Pilih **DeiT-Tiny** untuk aplikasi real-time dengan constraint waktu ketat. Dengan throughput 69.10 img/s dan akurasi 94.98%, model ini optimal untuk: video surveillance, autonomous vehicles, AR/VR applications, live streaming analysis.
- **Balanced Performance**: **ViT-Tiny** optimal untuk general-purpose classification dengan resource terbatas namun tetap memerlukan akurasi tinggi (95.64%). Cocok untuk: web applications, mobile deployment, edge computing dengan moderate constraints.
- **Class-Specific Deployment**:
  - Dataset dengan dominasi animal classes → gunakan Swin-Tiny untuk akurasi maksimal
  - Mixed objects dengan emphasis pada kecepatan → gunakan DeiT-Tiny
  - Application dengan balanced requirement → gunakan ViT-Tiny

5 KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil eksperimen perbandingan tiga model Vision Transformer (ViT-Tiny, Swin-Tiny, dan DeiT-Tiny) pada dataset CIFAR-10:

1. **Performa Keseluruhan**: Swin-Tiny mencapai akurasi tertinggi (96.25%) dengan margin 0.61% dari ViT-Tiny (95.64%) dan 1.27% dari DeiT-Tiny (94.98%). Semua model menunjukkan performa yang sangat baik dengan akurasi di atas 94%.
2. **Efisiensi Parameter**: ViT-Tiny dan DeiT-Tiny memiliki efisiensi parameter yang superior dengan 5.53M parameter (21.08 MB), sementara Swin-Tiny menggunakan 27.53M parameter (105.01 MB) untuk peningkatan akurasi yang relatif kecil.

3. **Kecepatan Inferensi**: DeiT-Tiny menunjukkan kecepatan inferensi terbaik (14.47 ms/image, 69.10 img/s), diikuti ViT-Tiny (69.36 ms/image, 14.42 img/s), dan Swin-Tiny (81.29 ms/image, 12.30 img/s).

4. **Trade-offs**: Terdapat trade-off yang jelas antara akurasi dan efisiensi komputasi. Swin-Tiny menawarkan akurasi tertinggi dengan cost parameter dan waktu inferensi yang lebih tinggi, sementara DeiT-Tiny memberikan efisiensi terbaik dengan akurasi yang masih kompetitif.

5.2 Rekomendasi

5.2.1 Akurasi Maksimal

Swin-Tiny: Untuk aplikasi medis, kontrol kualitas, penelitian akademis

5.2.2 Efisiensi Komputasi

ViT-Tiny/DeiT-Tiny: Mobile devices, edge computing, IoT devices

5.2.3 Aplikasi Real-time

DeiT-Tiny: Video surveillance, autonomous vehicles, AR applications

5.3 Saran Pengembangan

1. Eksplorasi model lain (BEiT, MAE, DINO)
2. Hyperparameter tuning dengan Bayesian optimization
3. Dataset lebih beragam
4. Quantization dan pruning
5. Attention visualization

6 DAFTAR PUSTAKA

1. Dosovitskiy, A., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv:2010.11929.
2. Liu, Z., et al. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. ICCV 2021.
3. Touvron, H., et al. (2021). Training data-efficient image transformers & distillation through attention. ICML 2021.
4. Vaswani, A., et al. (2017). Attention is all you need. NeurIPS 2017.
5. Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images.

7 LAMPIRAN

7.1 Source Code

Repository: https://github.com/Brammzz/VisionTransformer-Comparison
Berisi: Jupyter Notebook, Python scripts, requirements.txt, README, dokumentasi

7.2 Training Log

Training dengan early stopping berhasil converge dalam 3 epochs dengan memory optimization.

7.3 Hardware Specification

Hardware: CPU (Intel Core), Windows Machine
Software: Python 3.13+, PyTorch 2.0+, TIMM
Memory Optimization:

- Training pada CPU
- Batch size 8
- Garbage collection
- Reduced inference measurement