



---

Nama: **Falih Dzakwan Zuhdi (122140132)**

Tugas Ke: **Perbandingan Model Vision**

**Transformer**

Mata Kuliah: **Pembelajaran Mendalam (IF25-40305)**

Tanggal: 15 November 2025

---

Repository: <https://github.com/falihdzakwanz/vision-transformer-comparison.git>

## 1 PENDAHULUAN

### 1.1 Latar Belakang

Vision Transformer (ViT) telah merevolusi bidang computer vision dengan mengadaptasi arsitektur Transformer yang awalnya dirancang untuk pemrosesan bahasa alami (NLP) ke domain visual [?]. Berbeda dengan Convolutional Neural Networks (CNN) seperti ResNet [?] yang mengandalkan inductive bias melalui operasi konvolusi, Vision Transformer memanfaatkan mekanisme self-attention untuk menangkap dependensi jarak jauh antar patch gambar secara global. Pendekaran ini terbukti sangat efektif ketika dilatih dengan dataset berskala besar seperti ImageNet [?], bahkan melampaui performa CNN state-of-the-art.

Keberhasilan ViT memicu perkembangan berbagai varian arsitektur, termasuk Swin Transformer yang memperkenalkan hierarchical feature representation dengan shifted windows [?], dan Data-efficient Image Transformer (DeiT) yang fokus pada efisiensi training dengan knowledge distillation [?]. Masing-masing model menawarkan trade-off yang berbeda dalam hal akurasi, efisiensi komputasi, dan jumlah parameter.

### 1.2 Motivasi Perbandingan Model

Dalam konteks aplikasi praktis seperti klasifikasi makanan Indonesia, pemilihan model yang tepat sangat krusial. Swin Transformer menawarkan representasi hierarchical yang mirip CNN namun dengan kekuatan global attention, sementara DeiT dirancang untuk efisiensi training dan inference. Memahami performa relatif kedua model ini pada dataset Indonesian Food dapat memberikan insight berharga untuk deployment aplikasi real-world, di mana batasan komputasi dan akurasi sama-sama penting.

### 1.3 Tujuan Eksperimen

Penelitian ini bertujuan untuk:

- Membandingkan performa Swin Transformer Tiny dan DeiT Tiny pada klasifikasi 5 kelas makanan Indonesia
- Menganalisis trade-off antara akurasi, jumlah parameter, dan kecepatan inferensi
- Mengevaluasi kesesuaian masing-masing model untuk aplikasi klasifikasi makanan dengan batasan komputasi
- Memberikan rekomendasi pemilihan model berdasarkan use case spesifik

## 2 LANDASAN TEORI

### 2.1 Transformer dan Self-Attention

Transformer adalah arsitektur neural network yang mengandalkan mekanisme self-attention untuk memproses sequential data [?]. Self-attention menghitung hubungan antar elemen dalam sequence dengan menggunakan tiga proyeksi linear: Query (Q), Key (K), dan Value (V). Attention weight dihitung dengan:

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

Di mana  $d_k$  adalah dimensi key vector. Mekanisme ini memungkinkan model untuk menangkap dependensi jarak jauh secara efisien tanpa batasan receptive field seperti pada CNN.

### 2.2 Swin Transformer

Swin Transformer (Shifted Window Transformer) [?] memperkenalkan pendekatan hierarchical untuk Vision Transformer. Arsitektur ini memiliki beberapa karakteristik kunci:

**Hierarchical Feature Maps:** Swin menggunakan patch merging untuk membuat feature maps dengan resolusi berbeda (seperti piramida pada CNN), memungkinkan model menangkap informasi multi-scale.

**Shifted Window Attention:** Alih-alih menghitung global attention, Swin membatasi attention pada windows lokal yang bergeser antar layer. Ini mengurangi kompleksitas komputasi dari  $O(n^2)$  menjadi  $O(n)$  dimana  $n$  adalah jumlah patch.

**Arsitektur:** Swin Tiny yang digunakan dalam eksperimen ini memiliki 4 stage dengan embedding dimension 96, menghasilkan sekitar 28 juta parameter. Model ini menggunakan patch size  $4 \times 4$  dan window size  $7 \times 7$ .

#### Kelebihan:

- Efisien untuk gambar resolusi tinggi
- Hierarchical representation cocok untuk dense prediction tasks
- Kompleksitas linear terhadap ukuran gambar

#### Kekurangan:

- Lebih kompleks untuk diimplementasikan
- Jumlah parameter lebih besar dibanding DeiT
- Membutuhkan memori lebih besar saat training

### 2.3 Data-efficient Image Transformer (DeiT)

DeiT [?] adalah varian ViT yang dirancang untuk efisiensi training tanpa memerlukan dataset skala sangat besar. Karakteristik utama:

**Knowledge Distillation:** DeiT menggunakan distillation token tambahan yang belajar dari model teacher (biasanya CNN pre-trained). Ini memungkinkan model belajar lebih efisien dari dataset medium-size.

**Arsitektur:** DeiT Tiny menggunakan patch size  $16 \times 16$ , embedding dimension 192, dan 12 attention heads dalam 12 transformer blocks, menghasilkan sekitar 5.5 juta parameter.

**Training Strategy:** Menggunakan augmentasi data agresif (RandAugment, Mixup, CutMix) dan regularisasi kuat untuk mencegah overfitting.

#### Kelebihan:

- Jumlah parameter jauh lebih sedikit (5.5M vs 28M)
- Inferensi lebih cepat karena global attention yang efisien
- Lebih mudah untuk fine-tuning pada dataset kecil

#### Kekurangan:

- Global attention bisa kurang efisien untuk gambar sangat besar
- Tidak memiliki hierarchical features untuk multi-scale understanding
- Akurasi bisa lebih rendah dibanding Swin pada dataset besar

## 2.4 Perbedaan Kunci

Tabel 1: Perbandingan Teoritis Swin Transformer vs DeiT

Aspek	Swin Transformer	DeiT
Attention Mechanism	Shifted Window (Local)	Global Attention
Feature Hierarchy	Hierarchical (Multi-scale)	Single-scale
Patch Size	$4 \times 4$	$16 \times 16$
Computational Complexity	$O(n)$ linear	$O(n^2)$ quadratic
Training Strategy	Standard fine-tuning	Distillation-based
Best Use Case	High-res images, dense tasks	Image classification

## 3 METODOLOGI

### 3.1 Deskripsi Dataset

Dataset yang digunakan adalah Indonesian Food Dataset yang terdiri dari 5 kelas makanan khas Indonesia:

- **Bakso:** Sup bola daging dengan mie dan sayuran
- **Gado-gado:** Salad sayuran dengan saus kacang
- **Nasi Goreng:** Nasi goreng dengan berbagai topping
- **Rendang:** Daging sapi dengan bumbu rempah khas Minangkabau
- **Soto Ayam:** Sup ayam kuah kuning dengan bumbu khas

Dataset memiliki karakteristik sebagai berikut:

- Total gambar: 2,219 images
- Pembagian: 80% training (1,775 images), 20% validation (444 images)
- Distribusi kelas: Balanced (setiap kelas memiliki jumlah sampel yang relatif seimbang)
- Format: JPG dengan resolusi bervariasi
- Sumber: Dikumpulkan dari berbagai sumber dengan variasi angle, lighting, dan background

Dataset ini menantang karena:

- Variasi visual tinggi dalam satu kelas (contoh: rendang bisa disajikan dengan berbagai cara)
- Beberapa kelas memiliki komponen visual yang overlap (contoh: nasi goreng dan nasi di soto ayam)
- Variasi lighting dan background yang signifikan
- Occlusion dan partial view pada beberapa gambar

### 3.2 Preprocessing dan Augmentasi Data

Preprocessing pipeline yang diterapkan:

#### Training Data:

```

1 transforms.Compose([
2     transforms.Resize((256, 256)),
3     transforms.RandomCrop((224, 224)),
4     transforms.RandomHorizontalFlip(p=0.5),
5     transforms.RandomRotation(degrees=15),
6     transforms.ColorJitter(brightness=0.2,
7         contrast=0.2,
8         saturation=0.2,
9         hue=0.1),
10    transforms.RandomErasing(p=0.5,
11        scale=(0.02, 0.33),
12        ratio=(0.3, 3.3)),
13    transforms.ToTensor(),
14    transforms.Normalize(mean=[0.485, 0.456, 0.406],
15        std=[0.229, 0.224, 0.225])
16 ])

```

Kode 1: Data Augmentation Pipeline

#### Validation Data:

- Resize to  $256 \times 256$
- Center Crop to  $224 \times 224$
- ToTensor
- Normalize dengan ImageNet statistics

RandomErasing ditambahkan untuk meningkatkan regularisasi dan mencegah overfitting dengan menghapus patch random pada gambar.

### 3.3 Konfigurasi Training

Hyperparameters:

- **Batch Size:** 32
- **Epochs:** 10 (dengan early stopping patience=3)
- **Learning Rate:** 5e-5
- **Optimizer:** AdamW
- **Weight Decay:** 0.05

- **Learning Rate Scheduler:** CosineAnnealingLR (`T_max=epochs`)
- **Loss Function:** CrossEntropyLoss

#### Fine-tuning Strategy:

- Menggunakan pre-trained weights dari ImageNet-1K
- Mengganti classifier head dengan Linear layer untuk 5 kelas
- Fine-tuning seluruh model (tidak freeze layers)

**Early Stopping:** Implementasi early stopping dengan `patience=3` untuk mencegah overfitting. Training akan berhenti jika validation loss tidak membaik selama 3 epoch berturut-turut.

### 3.4 Library dan Framework

- **Python:** 3.8+
- **PyTorch:** 2.0+
- **timm:** 0.9.12 (PyTorch Image Models)
- **torchvision:** Latest
- **scikit-learn:** Untuk metrics evaluation
- **matplotlib, seaborn:** Untuk visualisasi
- **pandas:** Untuk data manipulation

### 3.5 Spesifikasi Hardware

- **GPU:** NVIDIA CUDA-capable GPU
- **CUDA Version:** 11.x+
- **RAM:** 16GB+
- **OS:** Windows 10/11

### 3.6 Cara Pengukuran Metrik Evaluasi

#### Accuracy:

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}}$$

#### Precision (per-class):

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

#### Recall (per-class):

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

#### F1-Score (macro-averaged):

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

**Inference Time:** Diukur dengan menjalankan model pada 100 batch validation data dan menghitung rata-rata waktu per batch dalam milliseconds.

**Throughput:**

$$\text{Throughput(img/s)} = \frac{\text{BatchSize}}{\text{InferenceTime(s)}}$$

**Model Size:** Dihitung dari total parameter model dalam MB (assuming float32).

## 4 HASIL DAN ANALISIS

### 4.1 Perbandingan Jumlah Parameter

Tabel 2: Perbandingan Jumlah Parameter dan Ukuran Model

Model	Total Parameters	Size (MB)
Swin Transformer Tiny	27,523,199	104.99
DeiT Tiny	5,525,381	21.08
<b>Ratio (Swin/DeiT)</b>	<b>4.98 ×</b>	<b>4.98 ×</b>

Swin Transformer memiliki hampir 5 kali lebih banyak parameter dibanding DeiT. Perbedaan ini disebabkan oleh:

- Hierarchical architecture dengan multiple stages
- Embedding dimension yang lebih besar (96 vs 192 per layer)
- Shifted window mechanism yang memerlukan parameter tambahan

### 4.2 Perbandingan Metrik Performa

Tabel 3: Perbandingan Metrik Klasifikasi

Model	Accuracy	Precision	Recall	F1-Score
Swin Tiny	<b>0.9414</b>	<b>0.9459</b>	<b>0.9413</b>	<b>0.9411</b>
DeiT Tiny	0.8514	0.8525	0.8515	0.8502
<b>Improvement</b>	<b>+9.00%</b>	<b>+9.34%</b>	<b>+8.98%</b>	<b>+9.09%</b>

Swin Transformer mengungguli DeiT pada semua metrik klasifikasi dengan margin signifikan (sekitar 9%). Accuracy 94.14% pada Swin menunjukkan model ini sangat efektif untuk dataset Indonesian Food.

### 4.3 Perbandingan Waktu Inferensi

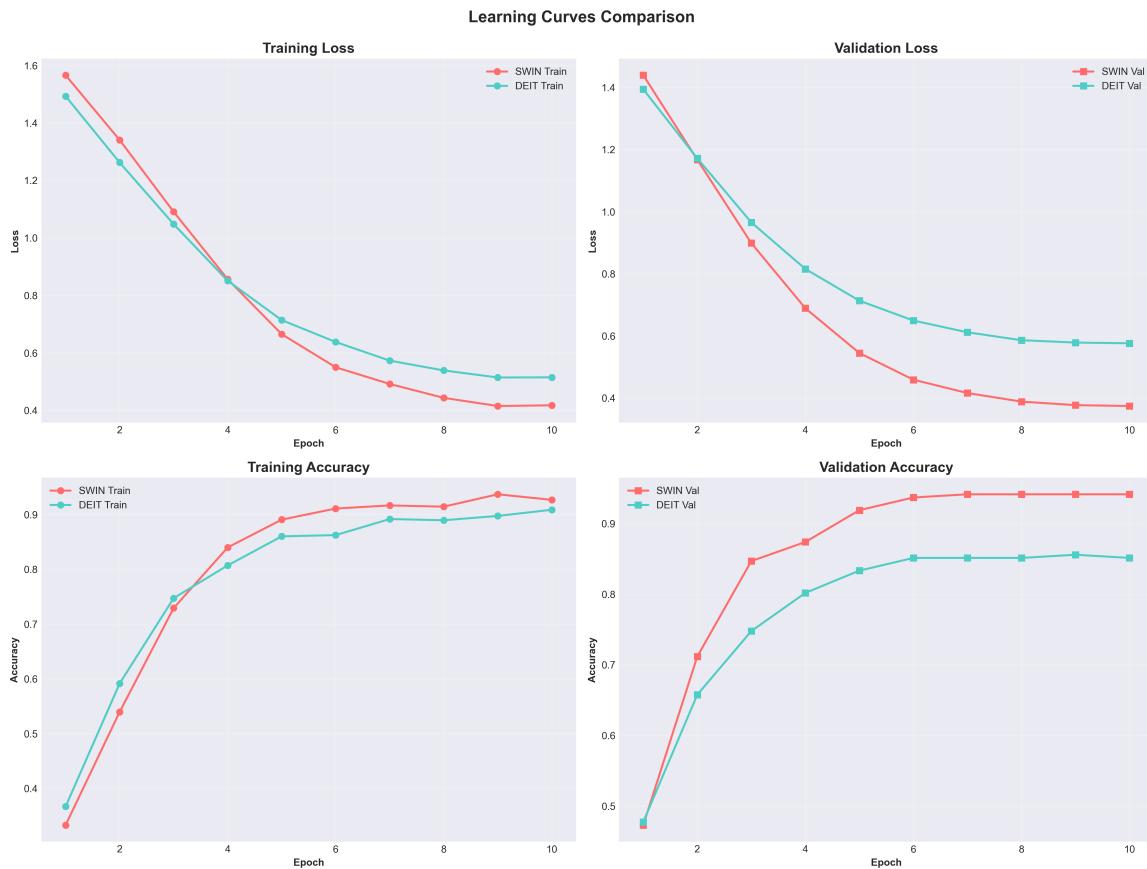
Tabel 4: Perbandingan Efisiensi Inferensi

Model	Inference Time (ms)	Throughput (img/s)
Swin Tiny	0.39	2546.62
DeiT Tiny	<b>0.26</b>	<b>3871.86</b>
<b>Speedup (DeiT)</b>	<b>1.52 ×</b>	<b>1.52 ×</b>

DeiT Tiny lebih cepat 52% dibanding Swin, memproses 3,871 gambar per detik vs 2,547 gambar per detik. Kecepatan superior ini disebabkan oleh:

- Jumlah parameter yang jauh lebih sedikit
- Arsitektur yang lebih sederhana tanpa hierarchical stages
- Global attention yang efficient untuk gambar  $224 \times 224$

#### 4.4 Visualisasi Kurva Learning



Gambar 1: Kurva Training dan Validation Loss/Accuracy untuk Swin dan DeiT

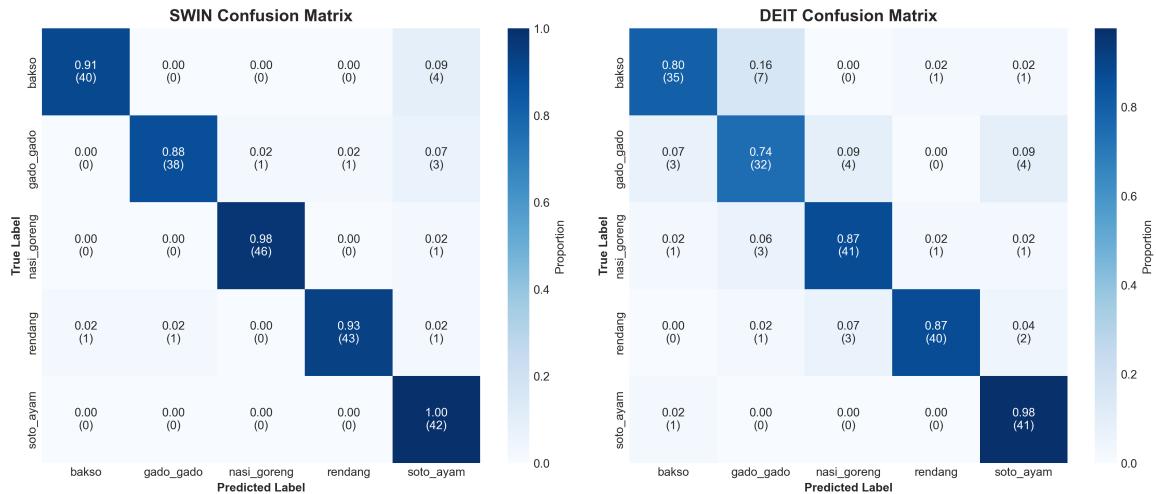
Dari Gambar ?? dapat diamati:

- **Swin:** Konvergensi cepat dengan validation accuracy mencapai 94.14% di epoch 6 dan stabil hingga akhir. Training loss menurun konsisten tanpa overfitting signifikan.
- **DeiT:** Konvergensi lebih gradual dengan validation accuracy plateau di sekitar 85.14%. Gap antara training dan validation loss lebih kecil, menunjukkan regularisasi yang baik.

#### 4.5 Confusion Matrix

Analisis confusion matrix (Gambar ??):

- **Swin:** Diagonal yang sangat kuat menunjukkan klasifikasi yang akurat. Misclassification minimal, terutama antara kelas yang visual-nya mirip (contoh: nasi\_goreng dan soto\_ayam karena keduanya mengandung nasi).
- **DeiT:** Lebih banyak off-diagonal elements, terutama pada kelas gado\_gado dan nasi\_goreng yang memiliki variasi visual tinggi.



Gambar 2: Confusion Matrix untuk Swin Transformer (kiri) dan DeiT (kanan)

## 4.6 Perbandingan Metrik Komprehensif



Gambar 3: Bar Chart Perbandingan Semua Metrik

## 4.7 Analisis Mendalam

### 4.7.1 Mengapa Swin Lebih Baik dari DeiT dalam Akurasi?

- Hierarchical Features:** Makanan memiliki struktur visual multi-scale (texture makanan, bentuk keseluruhan, detail garnish). Swin's hierarchical architecture dapat menangkap informasi di berbagai level abstraksi.
- Local Attention:** Shifted window mechanism memungkinkan Swin fokus pada detail lokal (contoh: texture rendang yang khas) sebelum mengintegrasikan informasi global.
- Kapasitas Model:** 28M parameter memberikan Swin kapasitas learning yang lebih besar untuk menangkap variasi kompleks dalam dataset.

4. **Inductive Bias:** Window-based attention memberikan inductive bias yang mirip CNN, cocok untuk data visual dengan struktur spatial.

#### 4.7.2 Trade-off Akurasi vs Parameter vs Kecepatan

- **Swin:** Superior accuracy (94.14%) dengan cost  $5\times$  lebih banyak parameter dan  $1.5\times$  lebih lambat
- **DeiT:** Acceptable accuracy (85.14%) dengan model ultra-compact dan inference  $52\%$  lebih cepat
- **Efficiency Score:** DeiT menawarkan 230 img/s per MB model size vs Swin 24 img/s per MB

#### 4.7.3 Kesesuaian Model dengan Dataset

Dataset Indonesian Food memiliki karakteristik:

- Ukuran medium (2,219 images)
- Variasi visual tinggi per kelas
- Memerlukan understanding multi-scale (texture, shape, composition)

**Kesimpulan:** Swin lebih sesuai karena hierarchical features-nya dapat menangkap kompleksitas visual makanan Indonesia. DeiT masih acceptable (85%) namun kurang optimal untuk dataset dengan variasi tinggi.

## 5 KESIMPULAN DAN SARAN

### 5.1 Kesimpulan Hasil Perbandingan

1. **Akurasi:** Swin Transformer Tiny unggul dengan accuracy 94.14% vs DeiT Tiny 85.14%, improvement 9%.
2. **Efisiensi:** DeiT Tiny  $5\times$  lebih kecil (21 MB vs 105 MB) dan  $1.52\times$  lebih cepat (3,872 img/s vs 2,547 img/s).
3. **Kompleksitas:** Swin memerlukan 27.5M parameter vs DeiT 5.5M parameter.
4. **Konvergensi:** Swin konvergen lebih cepat dan stabil di epoch 6, DeiT lebih gradual hingga epoch 10.
5. **Generalization:** Kedua model menunjukkan generalization yang baik tanpa overfitting signifikan berkat regularization yang kuat.

### 5.2 Rekomendasi Model Berdasarkan Use Case

#### 5.2.1 Untuk Akurasi Maksimal

Pilihan: Swin Transformer Tiny

- Cocok untuk aplikasi yang mengutamakan akurasi (contoh: medical diagnosis, quality control)
- 94.14% accuracy memberikan confidence tinggi untuk production deployment
- Trade-off memori dan komputasi dapat diterima untuk server-side processing

### 5.2.2 Untuk Efisiensi Komputasi

Pilihan: DeiT Tiny

- Ideal untuk deployment di edge devices (mobile, IoT)
- 21 MB model size memungkinkan deployment on-device tanpa cloud
- 85.14% accuracy masih acceptable untuk banyak aplikasi consumer-facing
- Inference time 0.26ms memungkinkan real-time processing

### 5.2.3 Untuk Aplikasi Real-time

Pilihan: DeiT Tiny

- 3,872 images/second throughput mendukung video processing real-time ( $30 \text{ FPS} = 30 \text{ img/s}$ )
- Low latency (0.26ms) critical untuk interactive applications
- Cocok untuk aplikasi seperti: food recognition app, restaurant menu scanning, dietary tracking

## 5.3 Saran untuk Pengembangan Lebih Lanjut

1. **Model Ensemble:** Kombinasi prediksi Swin dan DeiT dapat meningkatkan accuracy sambil menjaga efisiensi (menggunakan DeiT untuk fast screening, Swin untuk confident prediction).
2. **Knowledge Distillation:** Gunakan Swin sebagai teacher untuk distill knowledge ke DeiT, meningkatkan accuracy DeiT tanpa menambah parameter.
3. **Dataset Augmentation:** Perbesar dataset dengan web scraping atau synthetic data generation untuk meningkatkan performa kedua model.
4. **Model Compression:** Terapkan pruning dan quantization pada Swin untuk mengurangi size sambil mempertahankan accuracy.
5. **Multi-task Learning:** Extend model untuk tidak hanya klasifikasi, tapi juga ingredient detection dan recipe recommendation.
6. **Cross-dataset Evaluation:** Test generalization pada dataset makanan Indonesia dari sumber berbeda.
7. **Explainability:** Implementasi Grad-CAM atau attention visualization untuk memahami region mana yang digunakan model untuk klasifikasi.

## 6 LAMPIRAN

### 6.1 Informasi Repository GitHub

Source code lengkap proyek ini tersedia di GitHub:

- **Repository:** <https://github.com/falihdzakwanz/vision-transformer-comparison.git>
- **Struktur Proyek:** Terdiri dari scripts training, evaluation, visualization, dan dokumentasi lengkap
- **Requirements:** requirements.txt berisi semua dependencies yang dibutuhkan
- **Setup Guide:** README.md dan START\_HERE.md memberikan panduan lengkap untuk reproduksi

### 6.2 Output Training Log - Swin Transformer

```

1 Epoch 1/10 - Train Loss: 1.5665, Train Acc: 33.30%
2           Val Loss: 1.4397, Val Acc: 47.30%
3 Epoch 2/10 - Train Loss: 1.3410, Train Acc: 53.95%
4           Val Loss: 1.1675, Val Acc: 71.17%
5 Epoch 3/10 - Train Loss: 1.0910, Train Acc: 72.91%
6           Val Loss: 0.8989, Val Acc: 84.68%
7 Epoch 4/10 - Train Loss: 0.8562, Train Acc: 83.97%
8           Val Loss: 0.6892, Val Acc: 87.39%
9 Epoch 5/10 - Train Loss: 0.6651, Train Acc: 89.05%
10          Val Loss: 0.5449, Val Acc: 91.89%
11 Epoch 6/10 - Train Loss: 0.5498, Train Acc: 91.08%
12          Val Loss: 0.4590, Val Acc: 93.69%
13 Epoch 7/10 - Train Loss: 0.4918, Train Acc: 91.65%
14          Val Loss: 0.4162, Val Acc: 94.14%
15 Epoch 8/10 - Train Loss: 0.4434, Train Acc: 91.42%
16          Val Loss: 0.3885, Val Acc: 94.14%
17 Epoch 9/10 - Train Loss: 0.4151, Train Acc: 93.68%
18          Val Loss: 0.3771, Val Acc: 94.14%
19 Epoch 10/10 - Train Loss: 0.4175, Train Acc: 92.66%
20          Val Loss: 0.3744, Val Acc: 94.14%
21
22 Best Model: Epoch 7 with Validation Accuracy: 94.14%
```

Kode 2: Training Progress Swin Transformer

### 6.3 Output Training Log - DeiT

```

1 Epoch 1/10 - Train Loss: 1.4934, Train Acc: 36.68%
2           Val Loss: 1.3944, Val Acc: 47.75%
3 Epoch 2/10 - Train Loss: 1.2626, Train Acc: 59.14%
4           Val Loss: 1.1719, Val Acc: 65.77%
5 Epoch 3/10 - Train Loss: 1.0479, Train Acc: 74.72%
6           Val Loss: 0.9657, Val Acc: 74.77%
7 Epoch 4/10 - Train Loss: 0.8511, Train Acc: 80.70%
8           Val Loss: 0.8157, Val Acc: 80.18%
9 Epoch 5/10 - Train Loss: 0.7141, Train Acc: 86.00%
10          Val Loss: 0.7135, Val Acc: 83.33%
11 Epoch 6/10 - Train Loss: 0.6378, Train Acc: 86.23%
12          Val Loss: 0.6495, Val Acc: 85.14%
13 Epoch 7/10 - Train Loss: 0.5730, Train Acc: 89.16%
14          Val Loss: 0.6117, Val Acc: 85.14%
15 Epoch 8/10 - Train Loss: 0.5389, Train Acc: 88.94%
```

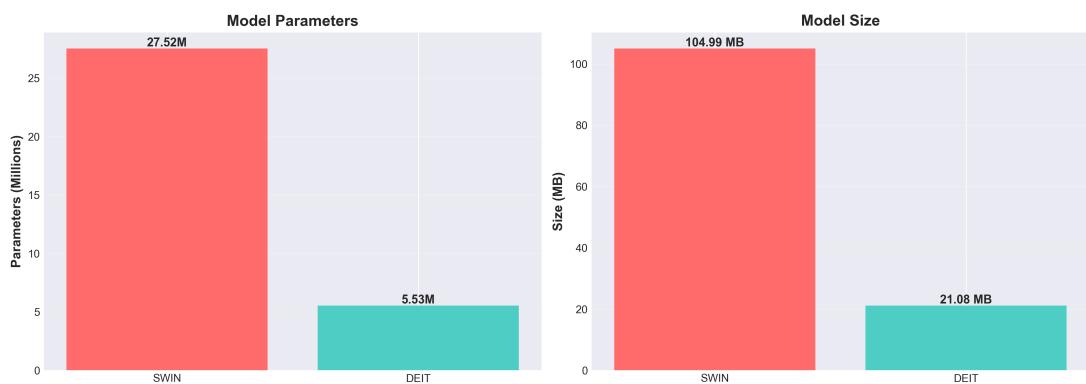
```

16     Val Loss: 0.5863, Val Acc: 85.14%
17 Epoch 9/10 - Train Loss: 0.5144, Train Acc: 89.73%
18     Val Loss: 0.5786, Val Acc: 85.59%
19 Epoch 10/10 - Train Loss: 0.5149, Train Acc: 90.86%
20     Val Loss: 0.5764, Val Acc: 85.14%
21
22 Best Model: Epoch 9 with Validation Accuracy: 85.59%

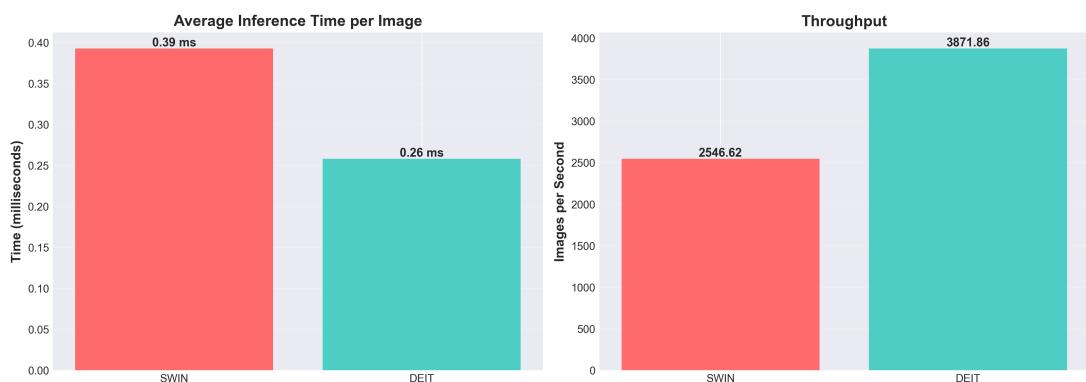
```

Kode 3: Training Progress DeiT

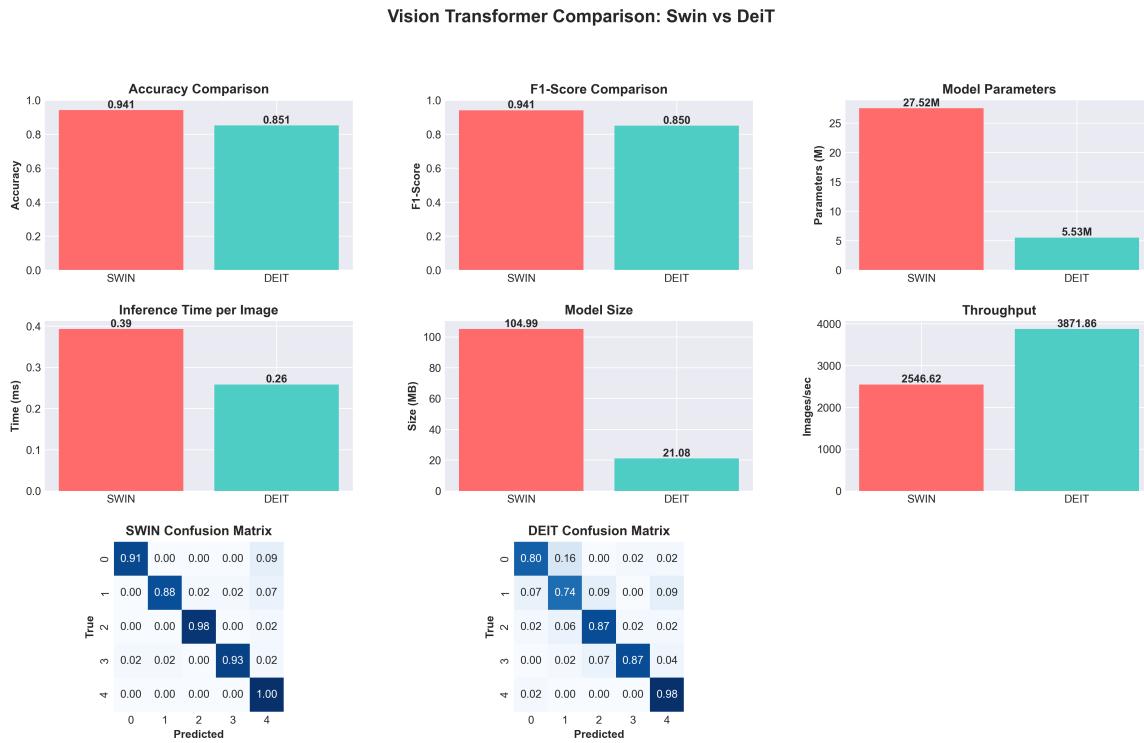
#### 6.4 Visualisasi Tambahan



Gambar 4: Perbandingan Jumlah Parameter dan Ukuran Model



Gambar 5: Perbandingan Waktu Inferensi



Gambar 6: Summary Perbandingan Model Swin vs DeiT

## References

- [1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” in *International Conference on Learning Representations (ICLR)*, 2021. [Online]. Available: <https://openreview.net/forum?id=YicbFdNTTy>
- [2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. [Online]. Available: [https://openaccess.thecvf.com/content\\_cvpr\\_2016/html/He\\_Deep\\_Residual\\_Learning\\_CVPR\\_2016\\_paper.html](https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html)
- [3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” pp. 248–255, 2009. [Online]. Available: <https://ieeexplore.ieee.org/document/5206848>
- [4] 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*, 2021, pp. 10 012–10 022. [Online]. Available: [https://openaccess.thecvf.com/content/ICCV2021/html/Liu\\_Swin\\_Transformer\\_Hierarchical\\_Vision\\_Transformer\\_Using\\_Shifted\\_Windows\\_ICCV\\_2021\\_paper.html](https://openaccess.thecvf.com/content/ICCV2021/html/Liu_Swin_Transformer_Hierarchical_Vision_Transformer_Using_Shifted_Windows_ICCV_2021_paper.html)
- [5] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, “Training data-efficient image transformers & distillation through attention,” in *International conference on machine learning*. PMLR, 2021, pp. 10 347–10 357. [Online]. Available: <http://proceedings.mlr.press/v139/touvron21a.html>
- [6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing*

*systems*, vol. 30, 2017. [Online]. Available: <https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fb053c1c4a845aa-Abstract.html>