Program Studi Teknik Informatika
Institut Teknologi Sumatera

Nama: **Bayu Ega Ferdana (122140129)** Tugas Ke: **Perbandingan Model Vision Transformer**
Mata Kuliah: **Pembelajaran Mendalam (IF25-40305)** Tanggal: 22 November 2025

Repository: https://github.com/Yuuggaa/deep-learning.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 [1]. Berbeda dengan Convolutional Neural Networks (CNN) seperti ResNet [2] yang mengandalkan inductive bias melalui operasi konvolusi, Vision Transformer memanfaatkan mekanisme self-attention untuk menangkap dependensi jarak jauh antar patch gambar secara global. Pendekatan ini terbukti sangat efektif ketika dilatih dengan dataset berskala besar seperti ImageNet [3], 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 [4]. 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 Vision Transformer (ViT) Base menggunakan pure transformer architecture dengan global attention mechanism. 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 Vision Transformer (ViT) Base 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 [5]. 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 Vision Transformer (ViT)

Vision Transformer [1] adalah implementasi murni dari Transformer architecture untuk computer vision. Karakteristik utama:

Patch Embedding: Gambar dibagi menjadi fixed-size patches (16×16), kemudian di-flatten dan diprojeksikan ke embedding space menggunakan linear layer.

Positional Encoding: Karena Transformer tidak memiliki inductive bias untuk spatial information, positional embeddings ditambahkan untuk memberikan informasi lokasi patch.

Global Self-Attention: Setiap patch dapat attend ke semua patch lainnya dalam gambar, memungkinkan model menangkap long-range dependencies.

Arsitektur ViT Base: Model yang digunakan memiliki:

- Patch size: 16×16
- Embedding dimension: 768
- Number of layers: 12
- Attention heads: 12
- Total parameters: 85.8 juta

Kelebihan:

- Dapat menangkap global context dengan efektif
- Scalable ke dataset sangat besar
- Transfer learning yang sangat baik dari ImageNet
- Architecture sederhana dan elegant

Kekurangan:

- Memerlukan dataset besar untuk training from scratch
- Computational complexity $O(n^2)$ untuk self-attention
- Kurang efisien untuk gambar beresolusi sangat tinggi
- Tidak memiliki hierarchical features seperti CNN

2.3 Swin Transformer

Swin Transformer (Shifted Window Transformer) [4] 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: Model yang digunakan memiliki:

- Patch size: 4×4
- Window size: 7×7
- Embedding dimension: 96
- Number of layers: 4 stages
- Total parameters: 27.5 juta

Kelebihan:

- Efisien untuk gambar resolusi tinggi (kompleksitas linear)
- Hierarchical representation cocok untuk dense prediction tasks
- Window-based attention memberikan inductive bias yang baik
- Balance antara local dan global information

Kekurangan:

- Arsitektur lebih kompleks dibanding pure ViT
- Shifted window mechanism menambah complexity implementasi
- Hyperparameter tuning lebih challenging (window size, shift size, dll)

2.4 Perbedaan Kunci

Tabel 1: Perbandingan Teoritis Swin Transformer vs ViT

Aspek	**Swin Transformer**	**ViT Base**
Attention Mechanism	Shifted Window (Local)	Global Attention
Feature Hierarchy	Hierarchical (Multi-scale)	Single-scale
Patch Size	4×4	16×16
Computational Complexity	O(n) linear	O(n^2) quadratic
Total Parameters	27.5M	85.8M
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: 1,108 images untuk training
- Format: JPG/JPEG dengan resolusi bervariasi
- Label: Disimpan dalam file CSV (train.csv)
- 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:

```
transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.RandomHorizontalFlip(),
    transforms.RandomRotation(10),
    transforms.ColorJitter(brightness=0.2, contrast=0.2),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406],
                         std=[0.229, 0.224, 0.225])
])
```

Kode 1: Data Augmentation Pipeline

Validation Data:

```
transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406],
                         std=[0.229, 0.224, 0.225])
])
```

Kode 2: Validation Transform

3.3 Konfigurasi Training

Hyperparameters:

- **Batch Size:** 32
- **Epochs:** 10
- **Learning Rate:** 1e-4
- **Optimizer:** AdamW
- **Weight Decay:** 0.01
- **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)

3.4 Library dan Framework

- **Python:** 3.8+
- **PyTorch:** 2.0+
- **timm:** 0.9.0+ (PyTorch Image Models)
- **torchvision:** Latest
- **scikit-learn:** Untuk metrics evaluation
- **matplotlib, seaborn:** Untuk visualisasi
- **pandas:** Untuk data manipulation

3.5 Spesifikasi Hardware

- **GPU:** NVIDIA GeForce RTX 3050 Laptop GPU
- **CUDA Version:** 11.x+
- **RAM:** 16GB
- **OS:** Windows 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:

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

Inference Time: Diukur dengan menjalankan model pada seluruh validation dataset dan menghitung rata-rata waktu per gambar dalam milliseconds, dengan warm-up run untuk memastikan GPU telah siap.

Throughput:

$$\text{Throughput (img/s)} = \frac{\text{Total Images}}{\text{Total Inference Time (s)}}$$

Model Size: Dihitung dari total bytes yang digunakan oleh parameter dan buffer model, dikonversi ke MB.

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	106.06
Vision Transformer Base	85,802,501	327.31
Ratio (ViT/Swin)	**3.12×**	**3.09×**

Vision Transformer Base memiliki lebih dari 3 kali lipat parameter dibanding Swin Transformer Tiny. Perbedaan ini disebabkan oleh:

- Embedding dimension yang lebih besar (768 vs 96)
- Jumlah transformer layers yang lebih banyak (12 vs 4 stages)
- Global attention mechanism yang memerlukan lebih banyak parameter
- Patch size yang lebih besar (16×16 vs 4×4) namun dengan embedding dimension yang jauh lebih tinggi

4.2 Perbandingan Metrik Performa

Hasil yang mengejutkan menunjukkan bahwa kedua model mencapai performa klasifikasi yang hampir identik dengan accuracy 99.91%. Ini mengindikasikan bahwa:

- Dataset Indonesian Food dengan 5 kelas ini relatif mudah untuk kedua arsitektur advanced ini
- Pre-trained weights dari ImageNet sangat efektif untuk transfer learning pada domain makanan
- Fine-tuning 10 epoch sudah cukup untuk mencapai konvergensi optimal

Tabel 3: Perbandingan Metrik Klasifikasi

Model	Accuracy	Precision	Recall	F1-Score
Swin Tiny	**99.91%**	**99.91%**	**99.91%**	**99.91%**
ViT Base	**99.91%**	**99.91%**	**99.91%**	**99.91%**
Difference	**0.00%**	**0.00%**	**0.00%**	**0.00%**

4.3 Perbandingan Waktu Inferensi

Tabel 4: Perbandingan Efisiensi Inferensi

Model	Time/Image (ms)	Throughput (img/s)	Hardware
Swin Tiny	**5.11**	**195.62**	RTX 3050
ViT Base	11.68	85.59	RTX 3050
Speedup (Swin)	**2.29×**	**2.29×**	-

Swin Transformer Tiny jauh lebih cepat (2.29× speedup) dibanding ViT Base dalam inference:

- 5.11 ms vs 11.68 ms per gambar
- 195.62 img/s vs 85.59 img/s throughput
- Keunggulan ini disebabkan oleh:
  - Jumlah parameter yang 3× lebih sedikit
  - Window-based attention dengan kompleksitas O(n) vs global attention $O(n^2)$
  - Hierarchical architecture yang lebih efisien untuk gambar 224×224

4.4 Visualisasi Kurva Learning

Dari analisis training curves dapat diamati:

Swin Transformer:

- Konvergensi sangat cepat, mencapai validation accuracy 99.19% di epoch 1
- Mencapai accuracy maksimal 100% di epoch 3 dan 4
- Training loss menurun konsisten dari 0.5276 → 0.0410
- Validation loss sangat rendah (0.0036 di epoch terakhir)
- Best model di epoch 3 dengan 100% validation accuracy

Vision Transformer Base:

- Konvergensi juga cepat, mencapai validation accuracy 91.88% di epoch 1
- Mencapai accuracy maksimal 100% di epoch 4, 5, dan 6
- Training loss menurun dari 0.5580 → 0.0350
- Beberapa fluktuasi di epoch 7-9 (val acc turun ke 96.12%)
- Best model di epoch 4 dengan 100% validation accuracy

4.5 Confusion Matrix

Analisis confusion matrix menunjukkan:

Swin Transformer:

- Diagonal sangat kuat dengan minimal misclassification
- Hanya 1 error: nasi_goreng diprediksi sebagai soto_ayam (1 dari 234 samples)
- Per-class accuracy hampir sempurna untuk semua kelas
- Classes dengan 100% accuracy: bakso, gado_gado, rendang, soto_ayam

Vision Transformer Base:

- Juga sangat akurat dengan minimal error
- 2 errors total:
  - 1 gado_gado diprediksi sebagai bakso
  - 1 rendang diprediksi sebagai bakso
- Classes dengan 100% accuracy: bakso, nasi_goreng, soto_ayam

4.6 Analisis Per-Class Metrics

Swin Transformer - Per Class:

Tabel 5: Swin Transformer - Metrik Per Kelas

Class	Precision	Recall	F1-Score	Support
bakso	100.00%	100.00%	100.00%	220
gado_gado	100.00%	100.00%	100.00%	215
nasi_goreng	99.57%	100.00%	99.79%	234
rendang	100.00%	100.00%	100.00%	227
soto_ayam	100.00%	99.53%	99.76%	212

Vision Transformer Base - Per Class:

Tabel 6: ViT Base - Metrik Per Kelas

Class	Precision	Recall	F1-Score	Support
bakso	100.00%	100.00%	100.00%	220
gado_gado	99.54%	100.00%	99.77%	215
nasi_goreng	100.00%	100.00%	100.00%	234
rendang	100.00%	99.56%	99.78%	227
soto_ayam	100.00%	100.00%	100.00%	212

4.7 Analisis Mendalam

4.7.1 Mengapa Kedua Model Mencapai Performa Hampir Identik?

1. **Dataset Characteristics:** Dataset Indonesian Food dengan 5 kelas dan 1,100 images tidak cukup kompleks untuk membedakan capabilities superior dari arsitektur yang berbeda. Kedua model sudah "overqualified" untuk task ini.

2. **Transfer Learning Power:** Pre-trained weights dari ImageNet-1K sangat powerful. Kedua model sudah belajar visual representations yang kaya, sehingga fine-tuning pada food dataset menjadi relatively straightforward.
3. **Class Separability:** 5 kelas makanan Indonesia ini memiliki visual features yang cukup distinct (bakso dengan kuah, gado-gado dengan sayuran, rendang dengan texture khas, dll), sehingga mudah untuk dipisahkan.
4. **Data Augmentation:** Augmentation yang kuat (flip, rotation, color jitter) membantu kedua model generalize dengan baik.

4.7.2 Trade-off Performa vs Efisiensi

Meskipun accuracy hampir identik, ada perbedaan signifikan dalam efisiensi:

Tabel 7: Trade-off Summary

Metric	Swin	ViT	Winner
Accuracy	99.91%	99.91%	Tie
Parameters	27.5M	85.8M	Swin (3.1×)
Model Size	106 MB	327 MB	Swin (3.1×)
Inference Speed	5.11 ms	11.68 ms	Swin (2.3×)
Throughput	195 img/s	85 img/s	Swin (2.3×)
Training Stability	Excellent	Good (fluctuations)	Swin

Kesimpulan Trade-off:

- **Swin Transformer** adalah clear winner untuk production deployment
- Memberikan accuracy yang sama dengan:
  - 3.1× lebih sedikit parameter (efisiensi memori)
  - 2.3× lebih cepat inference (critical untuk real-time apps)
  - Training yang lebih stabil
- **ViT Base** tidak memberikan keuntungan apapun meski 3× lebih besar

4.7.3 Efficiency Metrics

- **Swin Efficiency Score:** 1.84 img/s per MB (195.62 img/s ÷ 106 MB)
- **ViT Efficiency Score:** 0.26 img/s per MB (85.59 img/s ÷ 327 MB)
- **Swin is 7× more efficient** per MB of model size

4.7.4 Kapan ViT Base Bisa Lebih Unggul?

ViT Base mungkin menunjukkan keunggulan pada:

1. Dataset yang jauh lebih besar dan kompleks (100+ kelas, 100k+ images)
2. Tasks yang memerlukan very long-range dependencies
3. Domain yang sangat berbeda dari ImageNet (medical images, satellite imagery)
4. Fine-grained classification dengan subtle differences antar kelas

Namun untuk dataset Indonesian Food yang digunakan, **Swin Transformer Tiny is the optimal choice**.

5 KESIMPULAN DAN SARAN

5.1 Kesimpulan Hasil Perbandingan

1. **Akurasi:** Kedua model mencapai performa klasifikasi yang hampir identik (99.91% weighted average accuracy, precision, recall, dan F1-score).
2. **Efisiensi Parameter:** Swin Transformer Tiny jauh lebih efisien dengan 27.5M parameters (106 MB) dibandingkan ViT Base dengan 85.8M parameters (327 MB) - 3.1× lebih kecil.
3. **Kecepatan Inferensi:** Swin Transformer 2.3× lebih cepat (5.11 ms vs 11.68 ms per image, atau 195.62 img/s vs 85.59 img/s throughput).
4. **Konvergensi Training:** Swin Transformer menunjukkan training yang lebih stabil dengan konvergensi cepat ke 100% validation accuracy di epoch 3-4. ViT Base mengalami beberapa fluktuasi di epoch akhir.
5. **Overall Efficiency:** Swin Transformer 7× lebih efisien per MB model size (1.84 vs 0.26 img/s/MB).
6. **Hardware Compatibility:** Kedua model berjalan dengan baik pada NVIDIA RTX 3050 Laptop GPU, namun Swin memberikan throughput yang jauh lebih baik.

5.2 Rekomendasi Model Berdasarkan Use Case

5.2.1 Untuk Dataset Indonesian Food atau Similar Small-Medium Datasets

Rekomendasi Kuat: Swin Transformer Tiny

Alasan:

- Accuracy yang sama dengan ViT Base (99.91%)
- 3.1× lebih kecil - cocok untuk deployment di resource-constrained environments
- 2.3× lebih cepat - critical untuk real-time applications
- Training lebih stabil dan predictable
- Hierarchical features cocok untuk food classification (texture, color, shape di berbagai scales)

Use cases yang cocok:

- Mobile food recognition apps
- Restaurant menu scanning systems
- Dietary tracking applications
- Food delivery verification
- Nutritional analysis tools

5.2.2 Kapan ViT Base Bisa Dipertimbangkan

ViT Base hanya disarankan jika:

1. Dataset sangat besar (100k+ images, 100+ classes)
2. Computational resources tidak menjadi constraint (high-end server deployment)
3. Task memerlukan global understanding yang very sophisticated
4. Transfer learning dari ViT-specific pre-trained models diperlukan

Untuk Indonesian Food classification task, **ViT Base is overkill** dan tidak memberikan keuntungan apapun dibanding Swin Tiny.

5.2.3 Deployment Considerations

Tabel 8: Deployment Scenarios Recommendation

Scenario	Recommended Model	Reason
Mobile App	Swin Tiny	Small size, fast inference
Edge Device (IoT)	Swin Tiny	Memory efficient
Web API	Swin Tiny	Higher throughput
Server-side Batch	Swin Tiny	Same accuracy, lower cost
Research/Experiment	Either	Similar results

5.3 Saran untuk Pengembangan Lebih Lanjut

1. **Model Compression untuk Swin:**
   - Terapkan pruning untuk mengurangi parameter lebih lanjut tanpa mengorbankan accuracy
   - Quantization (FP32 $\rightarrow$ FP16/INT8) dapat meningkatkan speed 2-4$\times$
   - Knowledge distillation dari Swin Tiny ke model yang lebih kecil (Swin Pico)
2. **Dataset Expansion:**
   - Tambah kelas makanan Indonesia lainnya (sate, martabak, rawon, dll)
   - Perbesar dataset dengan web scraping atau synthetic data
   - Collect real-world data dengan berbagai kondisi lighting dan background
   - Test model pada dataset makanan dari negara lain untuk generalization study
3. **Multi-task Learning:**
   - Extend untuk ingredient detection
   - Recipe recommendation based on recognized food
   - Calorie estimation
   - Portion size detection
4. **Model Optimization Experiments:**
   - Try Swin Tiny with different window sizes

- Compare dengan Swin Base untuk melihat apakah extra parameters memberikan improvement
- Experiment dengan ViT Small/Tiny untuk fair comparison
- Test newer architectures (Swin V2, DINOv2, etc.)

5. **Production Deployment:**
   - Deploy Swin Tiny ke mobile app menggunakan TorchScript/ONNX
   - Implement A/B testing dengan real users
   - Monitor model drift dan retrain periodically
   - Implement explainability tools (Grad-CAM) untuk trust building
6. **Benchmark pada Hardware Lain:**
   - Test pada CPU untuk edge deployment scenarios
   - Benchmark pada mobile GPUs (Adreno, Mali)
   - Compare dengan specialized hardware (Google Coral TPU)
7. **Advanced Training Techniques:**
   - Implement progressive resizing (start 128×128, gradually increase to 224×224)
   - Try mixup/cutmix augmentation
   - Experiment dengan different learning rate schedules
   - Test test-time augmentation untuk boost accuracy lebih lanjut

5.4 Lesson Learned

1. **Bigger is NOT always better:** ViT Base dengan 3× lebih banyak parameter tidak memberikan accuracy gain apapun dibanding Swin Tiny pada dataset ini. Model size harus matched dengan task complexity.
2. **Efficiency matters in production:** Dalam deployment real-world, inference speed dan model size sama pentingnya dengan accuracy. Swin Tiny adalah optimal choice karena provides best balance.
3. **Transfer learning is powerful:** Kedua model pre-trained dari ImageNet sangat effective untuk food classification, menunjukkan visual features learned dari natural images transfer well ke food domain.
4. **Dataset size considerations:** Untuk dataset small-medium (1k images, 5 classes), advanced architectures seperti Swin Tiny atau ViT Base sudah sangat capable. Tidak perlu model yang lebih besar lagi.
5. **Hardware constraints:** Even pada mid-range laptop GPU (RTX 3050), kedua model berjalan dengan baik. Namun Swin Tiny memberikan better user experience dengan faster inference.

6 LAMPIRAN

6.1 Informasi Repository GitHub

Source code lengkap proyek ini tersedia di GitHub:

- **Repository:** https://github.com/Yuuggaa/deep-learning.git
- **Struktur Proyek:**
  - `python/`: Scripts untuk training, evaluation, dan visualization
  - `dataset/`: Folder untuk training data dan CSV labels
  - `models/`: Saved model checkpoints (.pth files)
  - `outputs/`: Hasil training logs, metrics, dan visualizations
  - `requirements.txt`: Python dependencies
- **Key Files:**
  - `train_swin.py`: Training script untuk Swin Transformer
  - `train_vit.py`: Training script untuk Vision Transformer
  - `evaluate.py`: Comprehensive evaluation dengan semua metrics
  - `visualize.py`: Generate learning curves
  - `model_swin.py`: Swin Transformer model definition
  - `model_vit.py`: ViT model definition
  - `dataset.py`: Custom dataset loader dengan CSV support

6.2 Output Training Log - Swin Transformer

```
MODEL INFORMATION: Swin Transformer
Total Parameters: 27,523,199
Trainable Parameters: 27,523,199
Non-trainable Parameters: 0
Model Size: 106.06 MB

Epoch 1/10 - Train Loss: 0.5276, Train Acc: 82.04%
Val Loss: 0.0389, Val Acc: 99.19%
Epoch 2/10 - Train Loss: 0.0863, Train Acc: 97.56%
Val Loss: 0.0154, Val Acc: 99.64%
Epoch 3/10 - Train Loss: 0.0216, Train Acc: 99.64%
Val Loss: 0.0030, Val Acc: 100.00%
Epoch 4/10 - Train Loss: 0.0066, Train Acc: 100.00%
Val Loss: 0.0007, Val Acc: 100.00%
Epoch 5/10 - Train Loss: 0.0069, Train Acc: 99.82%
Val Loss: 0.0020, Val Acc: 99.91%
Epoch 6/10 - Train Loss: 0.0088, Train Acc: 99.82%
Val Loss: 0.0043, Val Acc: 99.91%
Epoch 7/10 - Train Loss: 0.0215, Train Acc: 99.19%
Val Loss: 0.0040, Val Acc: 99.91%
Epoch 8/10 - Train Loss: 0.0332, Train Acc: 99.01%
Val Loss: 0.0017, Val Acc: 100.00%
Epoch 9/10 - Train Loss: 0.0336, Train Acc: 98.65%
Val Loss: 0.0276, Val Acc: 99.19%
Epoch 10/10 - Train Loss: 0.0410, Train Acc: 98.65%
Val Loss: 0.0036, Val Acc: 99.91%

Best Model: Epoch 3 with Validation Accuracy: 100.00%
```

Kode 3: Training Progress Swin Transformer

6.3 Output Training Log - Vision Transformer

```
MODEL INFORMATION: Vision Transformer (ViT)
Total Parameters: 85,802,501
Trainable Parameters: 85,802,501
Non-trainable Parameters: 0
Model Size: 327.31 MB

Epoch 1/10 - Train Loss: 0.5580, Train Acc: 80.69%
Val Loss: 0.2146, Val Acc: 91.88%
Epoch 2/10 - Train Loss: 0.1460, Train Acc: 94.31%
Val Loss: 0.0469, Val Acc: 98.65%
Epoch 3/10 - Train Loss: 0.0648, Train Acc: 97.56%
Val Loss: 0.0187, Val Acc: 99.46%
Epoch 4/10 - Train Loss: 0.0182, Train Acc: 99.37%
Val Loss: 0.0012, Val Acc: 100.00%
Epoch 5/10 - Train Loss: 0.0012, Train Acc: 100.00%
Val Loss: 0.0009, Val Acc: 100.00%
Epoch 6/10 - Train Loss: 0.0016, Train Acc: 99.91%
Val Loss: 0.0006, Val Acc: 100.00%
Epoch 7/10 - Train Loss: 0.0237, Train Acc: 99.37%
Val Loss: 0.1324, Val Acc: 96.12%
Epoch 8/10 - Train Loss: 0.0862, Train Acc: 97.29%
Val Loss: 0.0090, Val Acc: 99.55%
Epoch 9/10 - Train Loss: 0.1410, Train Acc: 95.94%
Val Loss: 0.0208, Val Acc: 99.64%
Epoch 10/10 - Train Loss: 0.0350, Train Acc: 98.74%
Val Loss: 0.0034, Val Acc: 99.91%

Best Model: Epoch 4 with Validation Accuracy: 100.00%
```

Kode 4: Training Progress Vision Transformer

6.4 Evaluation Results Summary

Swin Transformer Tiny:

```
EVALUATION RESULTS: Swin Transformer

B. PERFORMANCE METRICS
   Overall Accuracy: 99.91%

   Per-Class Metrics:
   bakso:      Precision: 100.00%, Recall: 100.00%, F1: 100.00%
   gado_gado:  Precision: 100.00%, Recall: 100.00%, F1: 100.00%
   nasi_goreng:Precision:  99.57%, Recall: 100.00%, F1:  99.79%
   rendang:    Precision: 100.00%, Recall: 100.00%, F1: 100.00%
   soto_ayam:  Precision: 100.00%, Recall:  99.53%, F1:  99.76%

   Weighted Average:
      Precision: 99.91%, Recall: 99.91%, F1-Score: 99.91%

C. INFERENCE TIME
   Average time per image: 5.11 ms
   Total time for 1108 images: 5.66 seconds
   Throughput: 195.62 images/second
   Hardware: GPU: NVIDIA GeForce RTX 3050 Laptop GPU
```

Kode 5: Swin Evaluation Summary

Vision Transformer Base:

```
EVALUATION RESULTS: Vision Transformer (ViT)

B. PERFORMANCE METRICS
   Overall Accuracy: 99.91%

   Per-Class Metrics:
   bakso:      Precision: 100.00%, Recall: 100.00%, F1: 100.00%
   gado_gado:  Precision:  99.54%, Recall: 100.00%, F1:  99.77%
   nasi_goreng:Precision: 100.00%, Recall: 100.00%, F1: 100.00%
   rendang:    Precision: 100.00%, Recall:  99.56%, F1:  99.78%
   soto_ayam:  Precision: 100.00%, Recall: 100.00%, F1: 100.00%

   Weighted Average:
      Precision: 99.91%, Recall: 99.91%, F1-Score: 99.91%

C. INFERENCE TIME
   Average time per image: 11.68 ms
   Total time for 1108 images: 12.95 seconds
   Throughput: 85.59 images/second
   Hardware: GPU: NVIDIA GeForce RTX 3050 Laptop GPU
```

Kode 6: ViT Evaluation Summary

6.5 Visualisasi Learning Curves

Visualisasi lengkap training dan validation curves untuk kedua model tersedia di:

- `outputs/figures/swin_training_curve.png`
- `outputs/figures/vit_training_curve.png`

Setiap visualisasi menampilkan 2 subplot:

- **Loss Curve:** Training loss (blue) dan Validation loss (red) per epoch
- **Accuracy Curve:** Training accuracy (blue) dan Validation accuracy (red) per epoch

6.6 Confusion Matrices

Confusion matrices untuk kedua model tersedia di:

- `outputs/figures/swin_cm.png`
- `outputs/figures/vit_cm.png`

Matrices menunjukkan distribusi prediksi untuk setiap kelas, dengan diagonal yang sangat kuat mengindikasikan high accuracy.

References

[1] 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," in *International Conference on Learning Representations (ICLR)*, 2021.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[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 (ICCV)*, 2021, pp. 10 012–10 022.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.