LAPORAN TUGAS EKSPLORASI

PERBANDINGAN MODEL VISION TRANSFORMER

(Swin Transformer, DeiT, dan MAE)

Disusun Oleh:

Rustian Afencius Marbun (122140155)

Mata Kuliah:

Pembelajaran Mendalam

Dosen Pengampu:

1. Imam Ekowicaksono, S.Si., M.Si.
2. Rahman Indra Kesuma, S.Kom., M.Cs.
3. Martin C.T.Manullang, P.hD.
4. Meida Cahyo Untoro, S.Kom., M.Kom.

PROGRAM STUDI TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI SUMATERA
2025

Link GitHub Repository:

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

DAFTAR ISI

1 PENDAHULUAN

1.1 Latar Belakang

Dalam beberapa tahun terakhir, bidang *Computer Vision* telah mengalami pergeseran paradigma yang signifikan dengan munculnya arsitektur Transformer. Awalnya dirancang untuk pemrosesan bahasa alami (*Natural Language Processing*/NLP), Transformer telah menunjukkan kemampuan luar biasa dalam memahami konteks global dalam citra, menantang dominasi *Convolutional Neural Networks* (CNN) yang telah lama menjadi standar industri [1, 2].

Vision Transformer (ViT) membagi citra menjadi *patch-patch* kecil dan memprosesnya sebagai urutan token, memungkinkan model untuk menangkap hubungan jarak jauh antar piksel yang seringkali sulit ditangkap oleh kernel konvolusi lokal. Namun, ViT standar memiliki tantangan tersendiri, seperti kebutuhan data yang sangat besar dan biaya komputasi yang tinggi. Hal ini memicu pengembangan berbagai varian ViT yang lebih efisien dan efektif.

1.2 Motivasi

Eksperimen ini dilatarbelakangi oleh urgensi untuk mengevaluasi karakteristik kinerja berbagai varian Vision Transformer (ViT) modern. Tiga model dipilih untuk merepresentasikan pendekatan berbeda dalam mengatasi keterbatasan ViT orisinal.

Pertama, Swin Transformer dipilih karena pendekatan uniknya yang menerapkan struktur hierarkis dan mekanisme *shifted windows*. Pendekatan ini dirancang untuk meningkatkan efisiensi komputasi dengan membatasi perhitungan *self-attention* pada jendela lokal, guna mengatasi masalah kompleksitas kuadratik yang terdapat pada ViT standar [3].

Selanjutnya, DeiT (Data-efficient Image Transformer) menjadi fokus analisis karena kemampuannya dalam menangani keterbatasan data pelatihan. Model ini menggunakan strategi distilasi pengetahuan (*knowledge distillation*) yang memungkinkan pelatihan model Transformer secara efisien dan berkinerja tinggi, meskipun menggunakan jumlah data yang lebih sedikit [4].

Terakhir, MAE (Masked Autoencoder) merepresentasikan pendekatan *self-supervised learning* yang inovatif. Melalui rekonstruksi bagian citra yang hilang (*masked*), MAE memungkinkan model untuk mempelajari representasi fitur yang kuat dan dapat digeneralisasi tanpa bergantung sepenuhnya pada data berlabel dalam jumlah besar [5].

1.3 Tujuan Eksperimen

Tujuan utama laporan ini adalah:

1. Membandingkan akurasi klasifikasi antara model Swin Transformer, DeiT, dan MAE pada dataset Flowers.
2. Menganalisis efisiensi komputasi, yang mencakup waktu inferensi dan *throughput*.
3. Mengevaluasi *trade-off* antara ukuran model (jumlah parameter) dan kinerjanya.
4. Memberikan rekomendasi pemilihan model berdasarkan berbagai skenario aplikasi.

2 LANDASAN TEORI

2.1 Transformer dan Self-Attention

Inti dari arsitektur Transformer adalah mekanisme *Self-Attention*, yang diperkenalkan oleh Vaswani et al. [6]. Mekanisme ini memungkinkan model untuk menimbang pentingnya setiap bagian input terhadap bagian lainnya. Secara matematis, *attention* didefinisikan sebagai:

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

Dimana Q (Query), K (Key), dan V (Value) adalah proyeksi dari input, dan d_k adalah dimensi dari key. Mekanisme ini memungkinkan pemodelan dependensi global tanpa batasan jarak spasial.

2.2 Deskripsi Model

2.2.1 Swin Transformer

Swin Transformer [3] memperkenalkan konsep *hierarchical feature maps* dan *shifted windows*. Berbeda dengan ViT yang memproses patch secara global, Swin membatasi *self-attention* pada jendela lokal yang tidak tumpang tindih, kemudian menggeser jendela tersebut pada layer berikutnya untuk memungkinkan komunikasi antar jendela. Ini menghasilkan kompleksitas komputasi yang linear terhadap ukuran citra, bukan kuadratik seperti pada ViT standar.

2.2.2 DeiT (Data-efficient Image Transformer)

DeiT [4] dirancang untuk mengatasi masalah kebutuhan data yang besar pada ViT. DeiT memperkenalkan token distilasi (*distillation token*) yang belajar dari output model

guru (biasanya CNN). Meskipun dalam eksperimen ini kita menggunakan arsitektur DeiT-Tiny, prinsip desainnya difokuskan pada efisiensi parameter dan kemampuan generalisasi yang baik pada dataset berukuran sedang.

2.2.3 MAE (Masked Autoencoder)

MAE [5] adalah pendekatan *self-supervised* di mana sebagian besar patch citra (misalnya 75%) ditutupi (*masked*), dan model dilatih untuk merekonstruksi piksel yang hilang. Encoder hanya memproses patch yang terlihat, membuatnya sangat efisien selama pre-training. Untuk tugas klasifikasi, encoder MAE (biasanya berbasis ViT-Base) di-*fine-tune* pada dataset target.

2.3 Perbedaan Kunci

Tabel 1: Perbedaan Karakteristik Model

Fitur	**Swin Transformer**	**DeiT**	**MAE (ViT-Base)**
Attention	Local (Windowed)	Global	Global
Struktur	Hierarkis	Kolumnar (Isotropik)	Kolumnar (Isotropik)
Fokus Utama	Efisiensi	Data Efficiency	Representation Learning
Ukuran	Tiny (dalam eksperimen)	Tiny	Base

2.4 Kelebihan dan Kekurangan Model

Setiap arsitektur model memiliki karakteristik unik yang memberikan keuntungan tertentu namun juga membawa keterbatasan. Berikut adalah analisis kelebihan dan kekurangan dari ketiga model yang dibandingkan.

Swin Transformer menawarkan kelebihan utama berupa efisiensi komputasi yang tinggi pada citra resolusi tinggi, berkat mekanisme *windowed attention* yang memiliki kompleksitas linear. Struktur hierarkisnya juga memungkinkan ekstraksi fitur multi-skala yang efektif. Namun, model ini memiliki kekurangan pada arsitekturnya yang lebih kompleks dibandingkan ViT standar, terutama pada implementasi *shifted windows* yang memerlukan manajemen memori yang lebih cermat.

DeiT (Data-efficient Image Transformer) unggul dalam kemampuannya mencapai performa tinggi dengan data pelatihan yang terbatas berkat strategi distilasi pengetahuan. Proses pelatihannya cenderung lebih stabil dan konvergen lebih cepat. Kekurangannya terletak pada ketergantungan kinerja model terhadap kualitas model guru (*teacher model*), serta kompleksitas tambahan dalam proses pelatihan yang melibatkan mekanisme distilasi.

MAE (Masked Autoencoder) memiliki keunggulan signifikan dalam efisiensi tahap *pre-training* karena hanya memproses sebagian kecil patch yang terlihat, serta kemampuannya mempelajari representasi fitur yang general tanpa label. Di sisi lain, kekurangannya adalah kebutuhan akan proses *fine-tuning* penuh pada seluruh parameter untuk tugas klasifikasi, yang dapat memakan sumber daya komputasi yang besar, terutama untuk varian model dengan ukuran *Base* atau lebih besar.

3 METODOLOGI

3.1 Dataset

Dataset yang digunakan adalah **Flowers Recognition Dataset** [7] yang memiliki total 3670 citra. Dataset ini terdiri dari 5 kelas bunga, yaitu Daisy, Dandelion, Rose, Sunflower, dan Tulip. Secara keseluruhan, dataset terbagi menjadi 2746 citra untuk data latih (*training*) dan 924 citra untuk data uji (*test*).

3.2 Preprocessing dan Augmentasi

Untuk meningkatkan generalisasi model, diterapkan beberapa teknik augmentasi data. Pertama, semua citra diubah ukurannya (*resize*) menjadi 224×224 piksel. Selanjutnya, dilakukan normalisasi menggunakan nilai mean $[0.485, 0.456, 0.406]$ dan standar deviasi $[0.229, 0.224, 0.225]$. Khusus untuk data latih, diterapkan augmentasi tambahan berupa *Random Horizontal Flip*, *Random Rotation* sebesar 15 derajat, *Color Jitter*, dan *Random Affine*.

3.3 Konfigurasi Training

Pelatihan dilakukan menggunakan *framework* PyTorch dan library `timm`. Berikut adalah konfigurasi hyperparameter yang digunakan.

Tabel 2: Konfigurasi Hyperparameter

Parameter	**Nilai**
Epochs	10
Learning Rate	1×10^{-4}
Optimizer	AdamW
Weight Decay	1×10^{-4}
Batch Size	32 (16 untuk Swin)
Loss Function	CrossEntropyLoss
Scheduler	Cosine Annealing

3.4 Library dan Framework

Implementasi eksperimen ini didukung oleh berbagai *library* dan *framework* Python sebagai berikut.

Tabel 3: Daftar Library dan Framework

Library/Framework	**Kegunaan**
PyTorch	Framework utama *deep learning*
timm	Penyedia model ViT *pre-trained*
Torchvision	Transformasi dan augmentasi citra
Scikit-learn	Perhitungan metrik evaluasi
Pandas	Manipulasi data tabular
Matplotlib & Seaborn	Visualisasi data dan hasil
NumPy	Operasi numerik dan array

3.5 Spesifikasi Hardware

Eksperimen ini dijalankan pada lingkungan pengembangan Jupyter Notebook di VS Code menggunakan perangkat laptop dengan spesifikasi prosesor Intel Core i5-13420H, GPU NVIDIA GeForce RTX 2050, dan RAM 16GB.

3.6 Pengukuran Metrik Evaluasi

Untuk mengukur kinerja model secara komprehensif, digunakan empat metrik evaluasi utama yaitu Akurasi, Precision, Recall, dan F1-Score. Perhitungan metrik ini didasarkan pada elemen-elemen dalam *confusion matrix*, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN).

Akurasi mengukur rasio prediksi yang benar terhadap total data:

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

Precision mengukur ketepatan model dalam memprediksi kelas positif:

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

Recall mengukur kemampuan model dalam menemukan kembali seluruh data kelas positif:

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

F1-Score adalah rata-rata harmonik dari Precision dan Recall, memberikan gam-

baran yang lebih seimbang jika terdapat ketimpangan kelas:

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

Selain itu, waktu inferensi diukur dalam milidetik (ms) per citra, dan *throughput* dihitung sebagai jumlah citra yang dapat diproses per detik.

4 HASIL DAN ANALISIS

4.1 Perbandingan Jumlah Parameter

Berikut adalah perbandingan jumlah parameter ketiga model.

Tabel 4: Perbandingan Ukuran Model

Model	**Parameter (Juta)**	**Ukuran (MB)**	**Tipe**
Swin Transformer	27.52	106.06	Tiny
DeiT	5.53	21.08	Tiny
MAE ViT	85.80	327.31	Base

4.2 Perbandingan Metrik Performa

Evaluasi dilakukan pada set validasi setelah 10 epoch pelatihan. Tabel berikut menampilkan performa model berdasarkan Akurasi, Precision, Recall, dan F1-Score.

Tabel 5: Perbandingan Metrik Evaluasi

Model	**Akurasi (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
Swin Transformer	**96.60**	**96.62**	**96.59**	**96.60**
DeiT	96.12	96.11	96.18	96.14
MAE ViT	95.39	95.50	95.25	95.34

Berdasarkan hasil evaluasi, **Swin Transformer** mencapai akurasi tertinggi sebesar 96.60%. Arsitektur hierarkisnya terbukti sangat efektif dalam menangkap fitur visual pada dataset bunga yang memiliki variasi bentuk dan tekstur yang kompleks. Hasil ini sejalan dengan temuan Wang [8] yang menyoroti efektivitas ViT dalam klasifikasi bunga. Sementara itu, **DeiT** menunjukkan performa yang sangat kompetitif dengan akurasi 96.12%, meskipun memiliki jumlah parameter yang jauh lebih sedikit (sekitar 5 kali lebih kecil dari Swin), yang membuktikan efisiensi arsitekturnya. Di sisi lain, **MAE ViT** mencatatkan akurasi terendah sebesar 95.39% dalam eksperimen ini. Hal ini kemungkinan disebabkan oleh ukuran model yang besar (ViT-Base) yang memerlukan

lebih banyak data atau durasi pelatihan yang lebih lama untuk mencapai konvergensi optimal dibandingkan varian Tiny.

4.3 Perbandingan Waktu Inferensi

Kecepatan inferensi diukur dalam milidetik per citra (ms/img) dan *throughput* (citra per detik).

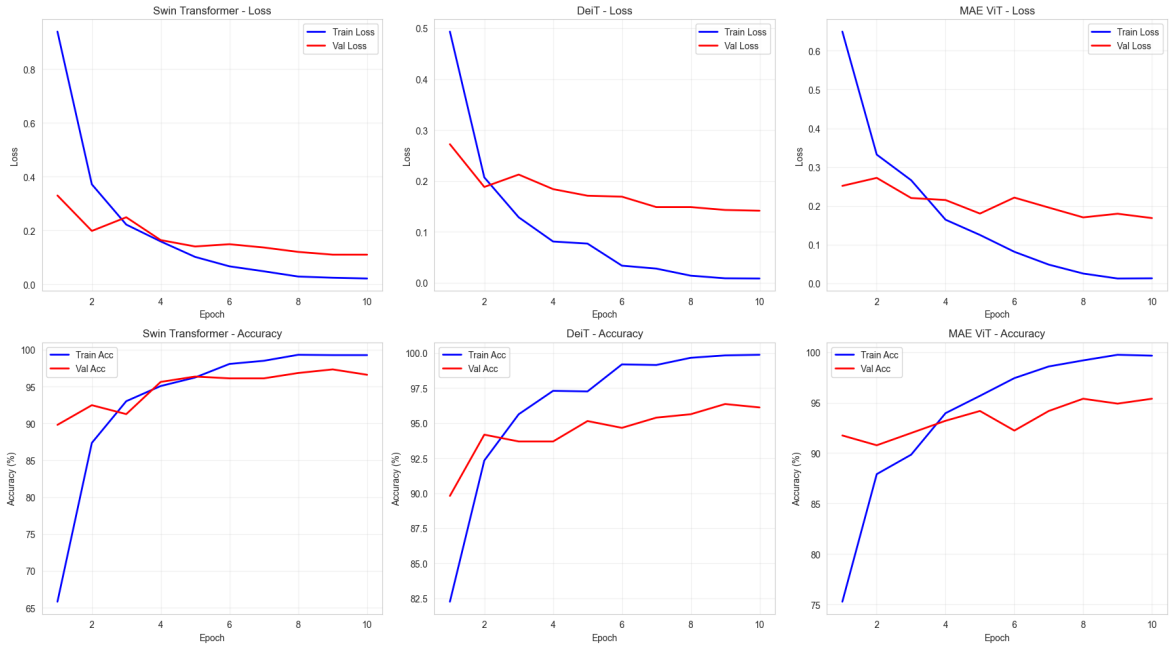
Tabel 6: Perbandingan Kecepatan Inferensi

Model	**Waktu (ms/img)**	**Throughput (img/s)**
Swin Transformer	**6.38**	**156.9**
DeiT	12.61	79.3
MAE ViT	73.03	13.7

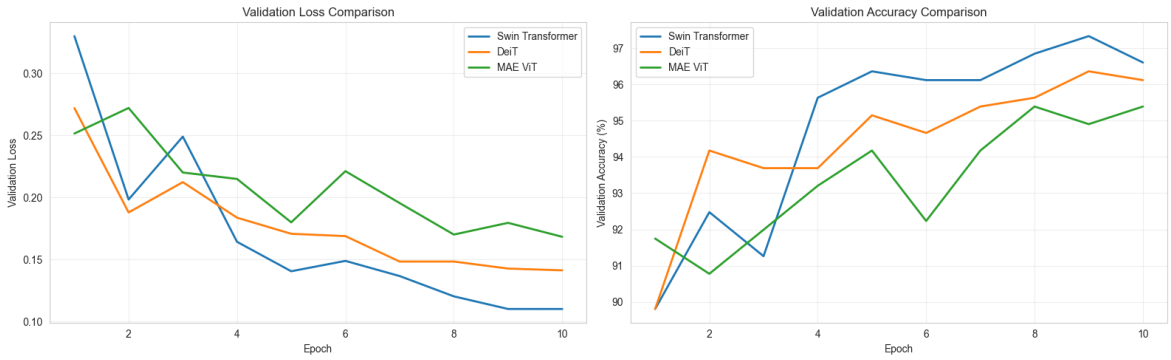
Dari segi kecepatan, **Swin Transformer** menjadi model tercepat dengan *throughput* mencapai 156.9 img/s. Hal ini dimungkinkan oleh mekanisme *windowed attention* yang membatasi komputasi hanya pada area lokal, sehingga mempercepat proses dibandingkan *global attention*. Sebaliknya, **MAE ViT** memiliki kinerja yang jauh lebih lambat dengan *throughput* hanya 13.7 img/s, yang disebabkan oleh kompleksitas kuadratik $O(N^2)$ dari mekanisme *global attention* pada resolusi penuh serta ukuran model yang besar.

4.4 Visualisasi Hasil

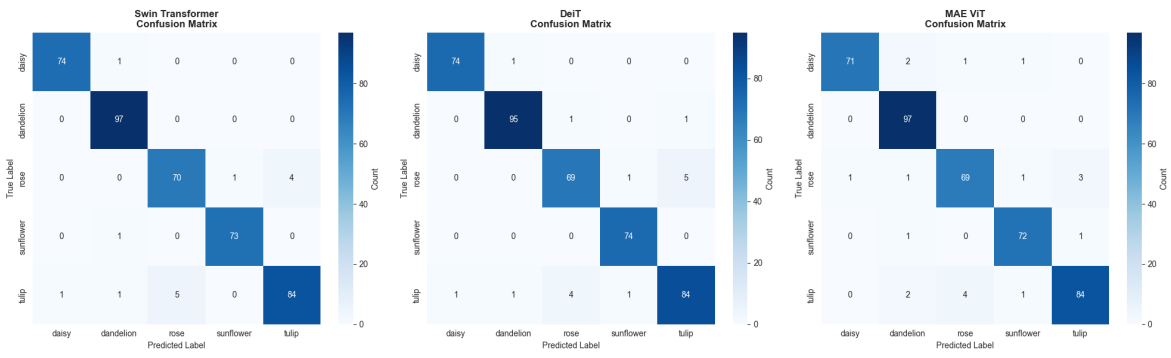
Berikut adalah visualisasi hasil pelatihan yang dihasilkan dari notebook, mencakup kurva loss dan confusion matrix.



(a) Grafik Perbandingan Training *Loss* dan *Accuracy*



(b) Grafik Perbandingan Validasi *Loss* dan *Accuracy*



(c) Confusion Matrix Perbandingan Model

Gambar 1: Visualisasi Hasil Eksperimen

4.5 Analisis Mendalam

4.5.1 Keunggulan Tiap Model

Berdasarkan hasil eksperimen, **Swin Transformer** menunjukkan keunggulan signifikan dibandingkan model lainnya, terutama dalam hal kecepatan inferensi dan akurasi. Keunggulan ini dapat diatribusikan pada mekanisme *shifted windows* yang membatasi perhitungan *self-attention* pada area lokal, mengurangi kompleksitas komputasi sekaligus menangkap detail tekstur bunga dengan lebih baik. Di sisi lain, **DeiT** unggul mutlak dalam aspek efisiensi parameter. Dengan hanya 5.5 juta parameter, DeiT mampu mendekati performa Swin (27.5 juta parameter), membuktikan efektivitas metode *knowledge distillation* dalam memadatkan pengetahuan model.

4.5.2 *Trade-off* Akurasi, Parameter, dan Kecepatan

- **Swin Transformer** menempati titik optimal untuk kinerja tinggi. Model ini menawarkan keseimbangan terbaik antara akurasi dan throughput tercepat, meskipun memiliki ukuran model yang moderat.
- **DeiT** menawarkan *trade-off* terbaik untuk efisiensi memori. Penurunan akurasi sebesar 0,48% dibandingkan Swin dikompensasi dengan pengurangan ukuran model yang signifikan sebesar 80%. Hal ini menjadikannya pilihan ideal untuk perangkat dengan sumber daya terbatas.
- **MAE ViT** menunjukkan *trade-off* yang kurang menguntungkan dalam eksperimen ini. Ukuran model yang besar (85 juta parameter) serta beban komputasi yang berat tidak sebanding dengan peningkatan akurasi, model ini justru menghasilkan kinerja terendah.

4.5.3 Kesesuaian Model dengan Dataset

- **DeiT** sangat sesuai dengan dataset ini karena desainnya yang spesifik untuk efisiensi data (*data-efficient*), memungkinkannya belajar dengan baik tanpa risiko *overfitting* yang besar.
- **Swin Transformer** juga menunjukkan kesesuaian yang tinggi, dimana struktur hierarkisnya membantu model menggeneralisasi fitur visual bunga dengan baik meskipun data terbatas.
- **MAE ViT** (Base) terlihat kurang sesuai untuk dataset skala ini jika ditinjau dari perspektif efisiensi *fine-tuning*. Model sebesar ini biasanya membutuhkan dataset yang jauh lebih masif untuk benar-benar memanfaatkan kapasitas pembelajarannya, sehingga cenderung sulit konvergen optimal pada dataset kecil.

5 KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil eksperimen perbandingan model Vision Transformer pada dataset Flowers, dapat disimpulkan beberapa poin penting sebagai berikut.

1. **Swin Transformer** keluar sebagai model dengan performa terbaik secara keseluruhan, mencatatkan akurasi tertinggi (96.60%) dan kecepatan inferensi tercepat.
2. **DeiT** membuktikan efisiensinya sebagai model yang sangat ringan (5.5M parameter) namun tetap mampu memberikan akurasi tinggi (96.12%).
3. **MAE ViT** (Base) kurang optimal untuk skenario ini dibandingkan varian Tiny lainnya, karena ukurannya yang besar membebani komputasi tanpa memberikan peningkatan akurasi yang signifikan pada dataset skala kecil ini.

5.2 Rekomendasi Pemilihan Model

Berdasarkan temuan eksperimen, untuk skenario yang membutuhkan akurasi maksimal dan pemrosesan *real-time*, disarankan untuk menggunakan **Swin Transformer** karena presisi tinggi dan respons cepatnya. Sedangkan untuk skenario yang mengutamakan efisiensi komputasi dan penggunaan pada perangkat edge, **DeiT** sangat direkomendasikan, terutama untuk aplikasi mobile atau IoT dengan keterbatasan memori.

5.3 Saran Pengembangan

Untuk pengembangan penelitian selanjutnya, disarankan untuk memperluas dataset dengan menambah jumlah sampel dan variasi kelas bunga guna menguji kemampuan generalisasi model secara lebih baik. Eksplorasi teknik augmentasi data yang lebih beragam serta penyetelan *hyperparameter* yang lebih mendalam juga diperlukan untuk memaksimalkan potensi setiap model. Selain itu, implementasi model ke dalam skenario penggunaan nyata, seperti aplikasi berbasis *mobile* atau web, dapat dilakukan untuk memvalidasi efektivitas model di luar lingkungan eksperimen.

Pustaka

[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," *arXiv preprint arXiv:2010.11929*, 2020.

[2] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu *et al.*, "A survey on vision transformer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 87–110, 2023.

[3] 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.

[4] 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.

[5] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.

[6] 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*, 2017, pp. 5998–6008.

[7] A. Mamaev, "Flowers recognition dataset," https://www.kaggle.com/alxmamaev/flowers-recognition, 2018, accessed: 2025-11-20.

[8] Y. Wang, "Flower classification and key parameter analysis based on vit," in *Proceedings of the 2024 International Conference on Image Processing and Media Computing (ICIPMC)*. SciTePress, 2024.

LAMPIRAN

A. Source Code

Source code lengkap eksperimen ini tersedia pada repository `https://github.com/username/reposit` dalam file Jupyter Notebook `vision_transformer_comparison.ipynb`.

B. Output Training Log

Log pelatihan lengkap tersimpan dalam repository github pada file `results/training_histories.jso`

C. Screenshot Hasil

Visualisasi hasil pelatihan telah dipindahkan ke Bab V (Hasil dan Analisis) untuk memudahkan pembacaan.