



Nama: **Freddy Harahap (122140018)**

Tugas: **Eksplorasi Vision Transformer**

Mata Kuliah: **Pembelajaran Mendalam (IF25-40401)**

Tanggal: 21 November 2025

1 Link Github

[Link GitHub](#)

2 Pendahuluan

Transformers menjadi fondasi utama dalam kemajuan *deep learning* berkat kemampuannya memodelkan hubungan dependensi jangka panjang melalui mekanisme *self-attention*, yang terbukti lebih efisien dan lebih mudah diparalelkan dibanding arsitektur berbasis *recurrent* maupun *convolutional* konvensional [1]. Keberhasilan arsitektur ini di bidang *NLP* mendorong adopsinya ke ranah *computer vision*, yang hingga beberapa tahun terakhir masih didominasi *CNN*. Perkembangan signifikan kemudian ditandai oleh *Vision Transformer (ViT)*, yang menunjukkan bahwa gambar dapat diproses sebagai urutan *patch* layaknya token teks dan tetap menghasilkan akurasi kompetitif ketika dilatih dalam skala besar [2]. Hal ini membuka paradigma baru bahwa representasi visual tidak harus bergantung pada *convolution* dan dapat sepenuhnya digantikan oleh *self-attention*.

Namun, efektivitas *ViT* membutuhkan skala dataset yang sangat besar, sehingga adopsinya dinilai kurang efisien untuk skenario data terbatas. Untuk mengatasi keterbatasan tersebut, *Data-efficient Image Transformer (DeiT)* dikembangkan dengan pendekatan pelatihan data-efficient dan teknik *distillation* berbasis token melalui atensi langsung dari *teacher model*, sehingga dapat mencapai performa tinggi meskipun hanya dilatih pada *ImageNet* tanpa dataset eksternal [3]. Pendekatan distilasi ini selaras dengan konsep *knowledge distillation*, yaitu mentransfer pengetahuan dari model besar ke model yang lebih kecil agar performanya tetap tinggi namun biaya komputasinya rendah [4]. Selain itu, beberapa penelitian menunjukkan bahwa strategi *transfer representation* dari pelatihan skala besar mampu meningkatkan efisiensi dan performa model di banyak *downstream task*, sehingga membandingkan arsitektur *Vision Transformer* dalam konteks efisiensi komputasi menjadi relevan baik untuk ranah riset maupun aplikasi industri [5].

Berdasarkan perkembangan tersebut, muncul kebutuhan evaluasi komparatif berbagai varian *Vision Transformer* untuk menentukan model yang paling sesuai bagi aplikasi dunia nyata. Sebab, kebutuhan *real-time* dan kebutuhan akurasi tinggi sering kali berlawanan: model besar umumnya memiliki akurasi lebih baik tetapi memiliki waktu inferensi lebih lambat, sedangkan model ringan dapat melakukan inferensi cepat namun dengan potensi penurunan akurasi. Evaluasi performa juga harus mempertimbangkan ukuran *parameter*, kebutuhan komputasi, dan *latency*, yaitu faktor yang menentukan kelayakan penerapan pada perangkat *edge* maupun *server-based*.

Penelitian ini bertujuan untuk:

1. membandingkan performa *ViT* dan *DeiT* pada dataset Indonesian Food sebagai representasi *fine-grained image classification*;
2. menganalisis hubungan jumlah parameter, akurasi, dan waktu inferensi untuk menilai efisiensi model; dan
3. memberikan rekomendasi model terbaik berdasarkan dua skenario aplikasi, yaitu *real-time* (prioritas inferensi cepat) dan *high-accuracy* (prioritas performa prediksi).

Melalui eksperimen komparatif ini, penelitian diharapkan memberikan panduan berbasis evidensi mengenai pemilihan arsitektur Vision Transformer yang tepat untuk implementasi praktis dan efisien di domain *computer vision*.

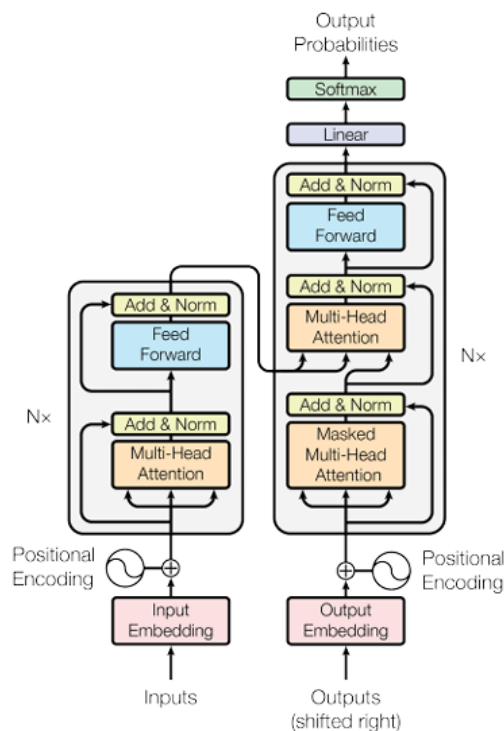
3 Landasan Teori

Bagian ini berisi kajian konsep yang digunakan dalam penelitian perbandingan kedua model.

3.1 Transformer dan Mekanisme Self-Attention

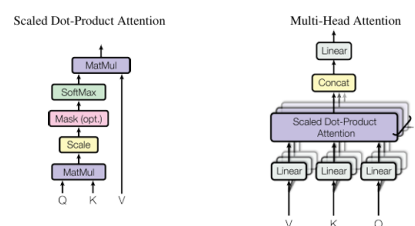
Transformer merupakan arsitektur pembelajaran mendalam yang dirancang untuk memodelkan dependensi jangka panjang tanpa bergantung pada struktur rekuren maupun konvolusi. Inti kinerjanya terletak pada mekanisme *self-attention*, yaitu proses ketika setiap token menilai dan merepresentasikan tingkat keterkaitannya terhadap token lain melalui perhitungan *query*, *key*, dan *value*, sehingga terbentuk representasi kontekstual yang kuat dalam satu tahap komputasi paralel. Pendekatan ini memiliki keunggulan berupa kemampuan paralelisasi yang tinggi, skalabilitas sesuai ukuran model, serta efisiensi dalam menangani relasi global pada data masukan.

Pada mekanisme *self-attention*, bobot perhatian dihitung menggunakan operasi *scaled dot-product* yang kemudian diproses dengan fungsi *softmax*. Perhitungan ini memungkinkan model memfokuskan perhatian pada bagian input yang paling relevan untuk tugas yang sedang diselesaikan. *Multi-Head Attention* memperluas konsep tersebut melalui penggunaan beberapa proyeksi perhatian yang berjalan secara independen sehingga model mampu menangkap pola dari berbagai subruang representasi. Dengan karakteristik tersebut, Transformer memformulasikan paradigma baru dalam *deep learning* dengan memanfaatkan konteks global secara efektif tanpa meningkatkan kompleksitas komputasi secara berlebihan [1].



Gambar 1: Arsitektur Original Transformer

Gambar 1 memperlihatkan arsitektur asli Transformer yang terdiri dari bagian *encoder* dan *decoder*. Pada sisi encoder, setiap token masukan diproyeksikan menjadi *embedding* kemudian digabungkan dengan *positional encoding* agar model tetap dapat mengenali urutan. Selanjutnya, token masukan diproses berulang oleh beberapa lapisan *Multi-Head Attention* dan *Feed-Forward Network* yang masing-masing dilengkapi *Add & Norm*. Pada sisi decoder, struktur serupa digunakan tetapi dengan penambahan *Masked Multi-Head Attention* untuk mencegah model melihat token masa depan selama proses pelatihan. Bagian decoder juga menerima keluaran encoder sebagai konteks untuk menghasilkan representasi yang kaya sebelum diproyeksikan ke distribusi probabilitas kata melalui lapisan *linear* dan *softmax*. Arsitektur ini menggambarkan karakter fundamental Transformer, yakni pembelajaran konteks global secara paralel tanpa ketergantungan berurutan seperti pada RNN.



Gambar 2: Proses Scaled Dot-Product dan Multi-Head Attention dalam Transformer

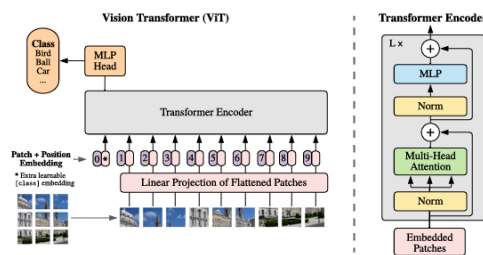
Secara lebih rinci, proses internal *self-attention* divisualisasikan pada Gambar 2. Panel kiri menunjukkan mekanisme *Scaled Dot-Product Attention*, di mana vektor *query* (Q), *key* (K), dan *value* (V) saling berinteraksi melalui operasi perkalian dot-product yang kemudian diskalakan dan diproses oleh fungsi *softmax* untuk menghasilkan bobot perhatian. Bobot ini kemudian digunakan untuk mengekstraksi representasi informasi paling relevan dari *value*. Panel kanan menunjukkan perluasan konsep ini menjadi *Multi-Head Attention*, yaitu menjalankan beberapa operasi perhatian secara paralel pada subruang

representasi yang berbeda. Setiap head menghasilkan keluaran perhatian independen, kemudian digabungkan (*concatenate*) dan diproyeksikan ulang melalui lapisan *linear*. Kombinasi multi-head ini memungkinkan model menangkap pola hubungan yang bervariasi dari token ke token dalam satu tahap komputasi, yang menjadi alasan inti keunggulan Transformer dalam memodelkan konteks global.

3.2 Arsitektur Vision Transformer (ViT)

Vision Transformer (ViT) mengadaptasi arsitektur Transformer ke ranah *computer vision* dengan memecah citra menjadi urutan *patch* berukuran tetap, kemudian merepresentasikan setiap patch sebagai vektor *embedding* serupa token pada tugas *NLP*. Seluruh patch diproses oleh susunan *Transformer Encoder* tanpa melibatkan operasi *convolution*. Prediksi akhir diperoleh dari *class token*, yaitu embedding tambahan yang disisipkan untuk menghimpun informasi global dari seluruh patch.

ViT menunjukkan performa akurasi yang kompetitif dibandingkan *CNN* ketika dilatih menggunakan dataset berskala besar seperti *ImageNet-21k* atau *JFT-300M*. Hal tersebut dimungkinkan karena mekanisme *self-attention* mampu mempelajari pola global secara menyeluruh pada citra. Namun, ViT bergantung kuat pada pelatihan berskala besar karena tidak memiliki *inductive bias* bawaan *CNN*, seperti *locality* dan *translation equivariance*, yang pada banyak kasus membantu proses generalisasi ketika dataset berukuran terbatas [2].



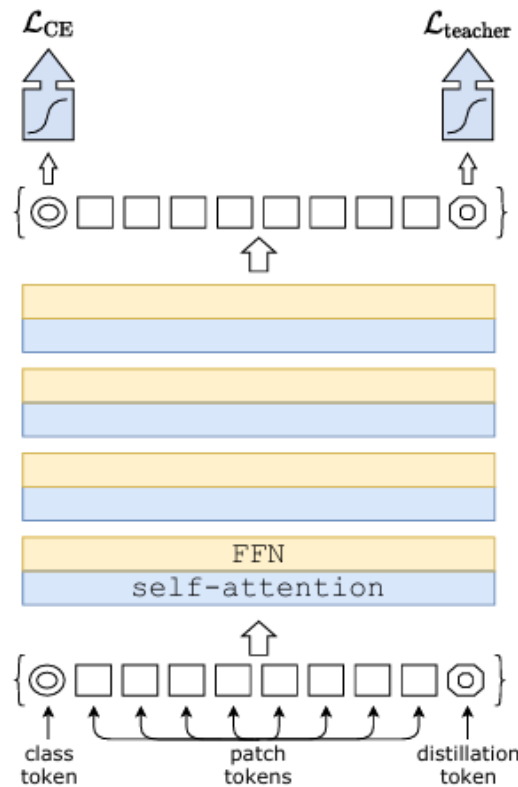
Gambar 3: Arsitektur Vision Transformer (ViT)

Gambar 3 menggambarkan alur pemrosesan pada *Vision Transformer*, dimulai dari proses pemecahan citra ke dalam bentuk patch berukuran tetap hingga tahap prediksi akhir melalui *class token*. Pada bagian bawah diagram terlihat kumpulan patch yang diproyeksikan menjadi *embedding* dan disusun sebagai urutan token masukan. Seluruh token tersebut kemudian melewati beberapa lapisan *Transformer Encoder* yang terdiri dari mekanisme *self-attention* dan *feed-forward network*. Setelah seluruh interaksi perhatian selesai dipelajari, hanya *class token* yang digunakan untuk menghasilkan prediksi akhir sehingga token tersebut bertindak sebagai agregator informasi global dari seluruh patch. Dengan demikian, gambar tersebut memperlihatkan karakteristik utama ViT, yaitu pemrosesan citra dalam bentuk urutan token secara penuh tanpa keterlibatan *convolution*.

3.3 Arsitektur DeiT (Data-Efficient Image Transformer)

DeiT dibangun berdasarkan struktur *ViT* namun memperkenalkan pendekatan pelatihan yang lebih efisien terhadap data. Kontribusi utamanya terletak pada mekanisme *token-based distillation*, yaitu proses pelatihan di mana sebuah *distillation token* berinteraksi melalui perhatian untuk menyerap pengetahuan dari *teacher model*. Dengan mekanisme tersebut, *student transformer* mampu mencapai akurasi kompetitif meskipun tidak dilatih menggunakan dataset berskala masif [3].

Secara prinsip, metode distilasi yang diterapkan pada *DeiT* selaras dengan kerangka teoritis *knowledge distillation*, yaitu transfer *soft targets* dari model berkapasitas besar ke model berkapasitas lebih kecil untuk meningkatkan kemampuan generalisasi, terutama dalam keterbatasan data. Pendekatan ini memungkinkan *DeiT* dipelajari secara efektif pada dataset berukuran terbatas dengan biaya komputasi yang jauh lebih rendah dibanding pelatihan *ViT* secara murni [4].



Gambar 4: Arsitektur Data-Efficient Image Transformer (DeiT)

Gambar 4 menggambarkan mekanisme *token-based distillation* pada *DeiT*. Pada tahap input terdapat tiga jenis token, yaitu *class token*, *patch tokens*, dan *distillation token*. Seluruh token diproses bersama melalui lapisan *self-attention* dan *feed-forward network* dalam *Transformer Encoder*. Setelah pemrosesan selesai, *class token* diarahkan ke *classification head* untuk menghitung *loss* utama \mathcal{L}_{CE} , sedangkan *distillation token* diarahkan ke *distillation head* untuk menghitung *loss* distilasi $\mathcal{L}_{teacher}$. Kedua *loss* dioptimasi secara simultan sehingga model belajar tidak hanya dari label dataset, tetapi juga dari *soft targets* yang diberikan *teacher model*.

3.4 Perbedaan ViT vs DeiT

Tabel 1: Perbandingan ViT dan DeiT berdasarkan aspek desain arsitektur

Aspek	ViT	DeiT
Kebutuhan data	Sangat tinggi; optimal pada dataset skala besar	Dirancang data-efficient pada dataset sedang/kecil
Komponen khusus	Tidak memiliki distillation token	Memiliki distillation token sebagai bagian pelatihan
Ketergantungan arsitektur	Self-attention murni	Self-attention + knowledge distillation
Tujuan desain	Akurasi maksimum saat pretraining skala besar	Akurasi optimal dengan sumber daya terbatas
Optimal pada	Industri skala besar / cloud inference	Deployment edge / limited compute

Selain itu, *ViT* mengandalkan transfer learning dari pretraining berskala besar untuk mencapai performa optimal, sedangkan *DeiT* berfokus pada *regularization-aware training* sehingga mampu memperoleh akurasi tinggi tanpa memerlukan pretraining masif.

3.5 Kelebihan dan Kekurangan Teoritis

3.5.1 Vision Transformer (ViT)

Kelebihan

- Representasi global diperoleh secara penuh melalui mekanisme *self-attention* tanpa kehilangan konteks lokal.
- Sangat mudah diskalakan ke ukuran model besar karena komputasi dapat diparalelkan secara penuh.
- Menunjukkan performa unggul pada dataset berskala sangat besar untuk visi komputer [2] serta strategi transfer skala besar seperti BiT [5].

Kekurangan

- Kinerja menurun signifikan pada dataset kecil karena tidak memiliki *inductive bias* khas *CNN*.
- Membutuhkan sumber daya pelatihan sangat tinggi dan dataset dalam skala jutaan contoh.

3.5.2 DeiT

Kelebihan

- Dapat mencapai akurasi mendekati atau bahkan melampaui *ViT* meskipun hanya menggunakan ImageNet-1K, berkat mekanisme *token-based distillation* [3].

- Lebih hemat komputasi dan lebih realistis untuk pelatihan maupun *deployment* pada sumber daya terbatas.
- Secara teoritis cocok untuk aplikasi yang membutuhkan inferensi cepat.

Kekurangan

- Performa sangat bergantung pada proses distilasi; tanpa *teacher model*, kinerjanya dapat menurun.
- Meskipun efisien, model masih dapat tertinggal dari *ViT* pada skala data yang sangat besar, di mana *ViT* umumnya tetap unggul.

4 Metodologi

Bagian ini menjelaskan tahapan eksperimen yang dilakukan dalam penelitian perbandingan model *Vision Transformer*, yang mencakup karakteristik dataset, strategi *preprocessing*, konfigurasi *training*, spesifikasi perangkat komputasi, serta metode evaluasi performa yang digunakan.

4.1 Dataset

Penelitian ini menggunakan dataset *Indonesian Food* yang berisi citra makanan khas Indonesia dalam format JPG. Dataset terdiri dari lima kelas, yaitu *bakso*, *gado-gado*, *nasi goreng*, *rendang*, dan *soto ayam*. Total jumlah kelas adalah lima dengan distribusi sampel yang relatif merata pada masing-masing kelas, sehingga sesuai untuk tugas klasifikasi multikelas. Dataset ini juga merupakan dataset yang telah digunakan dalam tugas Eksplorasi Resnet sebelumnya.

Dataset yang tersedia hanya mencakup berkas `train.csv` dan direktori `train/`, sedangkan `test.csv` tidak memiliki isi (kosong). Oleh karena itu, proses evaluasi tidak dapat mengandalkan *test set* terpisah. Untuk mengatasi keterbatasan tersebut, penelitian ini menerapkan skema *5-Fold Stratified Cross-Validation*, di mana setiap fold memiliki subset data latih dan subset data validasi. Pada setiap fold, subset validasi diperlakukan sebagai *test set* untuk perhitungan metrik evaluasi performa dan waktu inferensi.

4.2 *Preprocessing* dan Augmentasi

Seluruh citra diubah dari format RGB ke tensor dan di-*resize* menjadi ukuran 224×224 piksel sesuai standar masukan arsitektur *Vision Transformer*. Normalisasi dilakukan menggunakan parameter dataset ImageNet, dengan nilai *mean* $[0.485, 0.456, 0.406]$ dan *standard deviation* $[0.229, 0.224, 0.225]$.

Untuk meningkatkan kemampuan generalisasi model, augmentasi hanya diterapkan pada data latih menggunakan:

- *Horizontal Flip* ($p = 0.5$),
- *Random Brightness Contrast*,
- *Shift Scale Rotate*.

Sementara itu, data validasi tidak diberikan augmentasi tambahan selain *resize* dan normalisasi untuk menjaga kemurnian evaluasi.

4.3 Konfigurasi Training

Eksperimen dilakukan dengan pendekatan *fine-tuning* menggunakan bobot pra-latih (*pretrained weights*) ImageNet. Konfigurasi pelatihan diatur sama untuk kedua model agar perbandingan tetap adil (*fair comparison*). Rincian konfigurasi pelatihan ditunjukkan pada Tabel 2.

Tabel 2: Konfigurasi pelatihan yang digunakan pada eksperimen

Parameter	Nilai
Optimizer	AdamW
Learning Rate	1×10^{-4}
Weight Decay	1×10^{-4}
Batch Size	32
Epochs per Fold	5
Loss Function	Cross-Entropy Loss

Model yang dibandingkan pada eksperimen ini adalah sebagai berikut:

1. Vision Transformer Base Patch16 224 (ViT-B/16)
2. Data-Efficient Image Transformer Small (DeiT-Small / Patch16 224)

Kedua model menggunakan arsitektur *patch embedding* dan *multi-head self-attention*, dengan perbedaan utama pada efisiensi jumlah parameter serta keberadaan *data-distillation mechanism* pada *DeiT* yang tidak dimiliki *ViT*.

4.4 Library dan Framework

Eksperimen dilaksanakan pada platform Google Colab dengan dukungan pustaka sebagai berikut:

- PyTorch untuk implementasi *deep learning*;
- TIMM (PyTorch Image Models) untuk arsitektur *Vision Transformer*;
- Albumentations untuk augmentasi citra;
- Scikit-learn untuk *cross-validation* dan metrik evaluasi;
- Torchinfo untuk perhitungan parameter model;
- NumPy, Pandas, Matplotlib, dan Seaborn untuk analisis dan visualisasi.

4.5 Spesifikasi Hardware

Proses pelatihan dan evaluasi dijalankan pada perangkat komputasi dengan spesifikasi sebagai berikut:

- Platform: Google Colab
- GPU: NVIDIA Tesla T4 (16 GB)
- CPU: Intel Xeon virtual core
- RAM: ± 12 GB

Pemrosesan GPU digunakan untuk seluruh proses *forward pass*, *backpropagation*, dan *inference*.

4.6 Metode Evaluasi

Evaluasi dilakukan menggunakan skema 5-Fold Stratified Cross-Validation. Untuk setiap fold:

1. Training dijalankan pada subset data latih.
2. Validation/Test dijalankan pada subset data validasi (bertindak sebagai test set fold).
3. Model terbaik pada fold ditentukan berdasarkan akurasi tertinggi pada subset validasi.

Metrik evaluasi yang digunakan meliputi:

- Accuracy
- Precision, Recall, dan F1-Score (macro average dan per kelas)
- Confusion Matrix
- Kurva Training vs Validation Loss dan Accuracy

Selain metrik performa klasifikasi, penelitian juga mengukur efisiensi model melalui waktu inferensi dengan dua indikator:

1. Rata-rata waktu inferensi per citra (ms/img)
2. Throughput inferensi (gambar per detik / img/s)

Pengukuran inferensi dilakukan pada seluruh data validasi dalam keadaan *eval mode*, serta dilakukan *CUDA synchronization* untuk akurasi timing pada GPU.

5 Hasil dan Analisis

Bab ini menyajikan hasil eksperimen perbandingan kedua model pada dataset citra makanan Indonesia, serta analisis kritis terhadap hubungan antara jumlah parameter, performa klasifikasi, dan efisiensi waktu inferensi.

5.1 Hasil Kuantitatif

5.1.1 Perbandingan Jumlah Parameter

Tabel 3 menunjukkan perbandingan jumlah parameter antara ViT dan DeiT yang digunakan dalam eksperimen ini (berdasarkan hasil fungsi `count_parameters`).

Tabel 3: Perbandingan jumlah parameter model

Model	Total Params	Trainable Params	Non-trainable Params	Model Size (MB)
ViT	85.802.501	85.802.501	0	327,31
DeiT	21.667.589	21.667.589	0	82,66

Seluruh parameter pada kedua model dibuat trainable (tidak ada parameter yang dibekukan), sehingga perbandingan dilakukan pada konfigurasi full fine-tuning. Dari tabel terlihat bahwa DeiT hanya memiliki sekitar 25% parameter ViT (pengurangan kapasitas sekitar 75%), dengan ukuran model sekitar empat kali lebih kecil.

5.1.2 Metrik Performa Klasifikasi

Tabel 4 menampilkan performa rata-rata kedua model berdasarkan skema 5-fold stratified cross-validation. Nilai yang disajikan merupakan rata-rata lima fold untuk akurasi dan metrik makro (macro average).

Tabel 4: Perbandingan performa klasifikasi (rata-rata 5-fold)

Model	Val Acc (mean 5-fold)	Macro Precision (mean)	Macro Recall (mean)	Macro F1 (mean)
ViT	0.9597	0.9621	0.9596	0.9598
DeiT	0.9747	0.9756	0.9746	0.9747

Secara konsisten, DeiT memberikan akurasi dan F1-Score makro sedikit lebih tinggi dibandingkan ViT, dengan selisih sekitar 1,5–1,7 poin persentase. Artinya, meskipun kapasitas parameter DeiT jauh lebih kecil, kualitas klasifikasinya pada dataset ini tidak menurun, bahkan sedikit lebih baik.

Selain rata-rata 5-fold, pada fold terbaik (fold ke-4) diperoleh:

- ViT: akurasi validasi = 0.9784, macro F1 \approx 0.9784
- DeiT: akurasi validasi = 0.9838, macro F1 \approx 0.9838

Hasil tersebut kembali mengonfirmasi bahwa performa DeiT tetap konsisten dan kompetitif meskipun jumlah parameternya jauh lebih kecil dibandingkan ViT.

5.1.3 Waktu Inferensi

Tabel 5 memperlihatkan perbandingan efisiensi inferensi kedua model, dihitung pada subset validasi masing-masing fold dan kemudian dirata-rata.

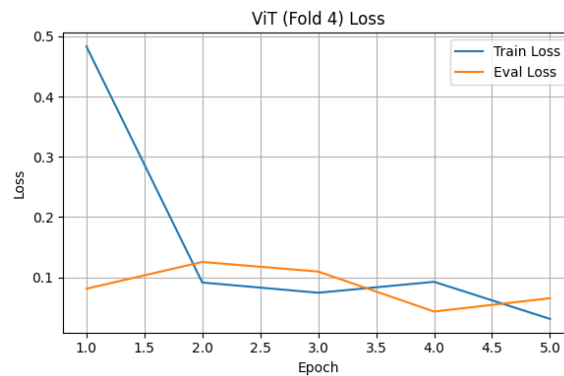
Tabel 5: Perbandingan waktu inferensi (rata-rata 5-fold)

Model	Avg Time (ms/img)	Throughput (img/s)
ViT	10,2144	97,90
DeiT	2,8262	353,93

DeiT membutuhkan rata-rata sekitar 2,83 ms per citra, sedangkan ViT sekitar 10,21 ms per citra. Dengan kata lain, ViT sekitar 3,6 kali lebih lambat, atau dari sudut pandang throughput, DeiT sekitar 3,6 kali lebih cepat memproses citra dibanding ViT pada konfigurasi hardware yang sama.

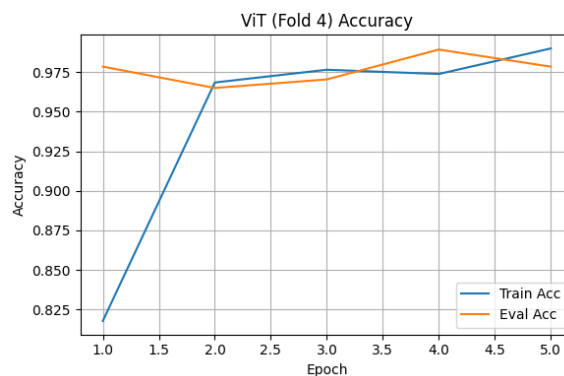
5.2 Visualisasi Kurva Belajar

Visualisasi kurva belajar digunakan untuk mengidentifikasi dinamika optimisasi yang terjadi selama proses pelatihan pada fold terbaik.



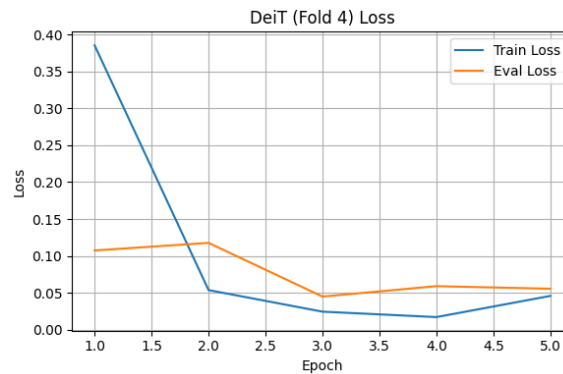
Gambar 5: Kurva training loss dan validation loss model ViT

Gambar 5 menunjukkan tren training loss dan validation loss pada model ViT. Training loss menurun tajam pada epoch pertama dan terus menurun hingga akhir pelatihan, sedangkan validation loss bertahan pada nilai rendah dengan sedikit fluktuasi pada epoch ke-4. Pola ini menunjukkan bahwa ViT mampu mempelajari pola dataset dengan cepat, tetapi terdapat perbedaan kecil antara training loss dan validation loss yang mengindikasikan gejala overfitting ringan.



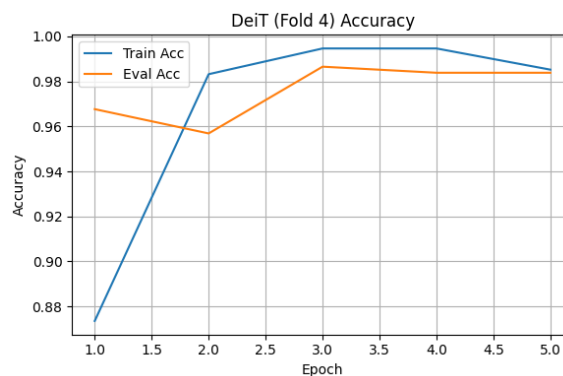
Gambar 6: Kurva training accuracy dan validation accuracy model ViT

Gambar 6 memperlihatkan bahwa training accuracy meningkat hampir linier hingga mendekati 1,00, sedangkan validation accuracy bergerak pada rentang 0,96–0,98. Hal ini konsisten dengan temuan sebelumnya bahwa kapasitas besar ViT memungkinkan model dengan cepat mencapai akurasi tinggi pada data latih, namun peningkatan kinerja pada data validasi tidak sebanding.



Gambar 7: Kurva training loss dan validation loss model DeiT

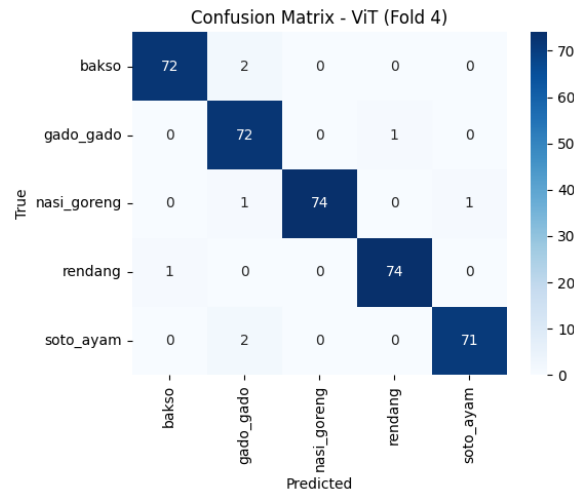
Untuk DeiT, kurva loss pada Gambar 7 menunjukkan penurunan yang cepat dan lebih stabil dibanding ViT. Training loss dan validation loss memiliki bentuk kurva yang halus dan saling mengikuti sepanjang pelatihan, menunjukkan generalization gap yang kecil serta proses optimisasi yang lebih stabil.



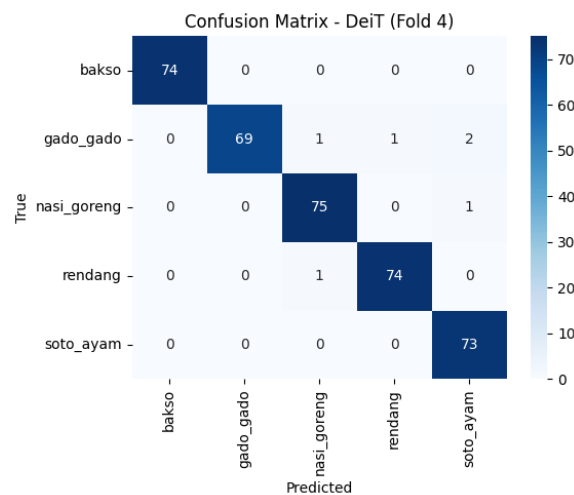
Gambar 8: Kurva training accuracy dan validation accuracy model DeiT

Gambar 8 menunjukkan bahwa training accuracy dan validation accuracy pada DeiT meningkat secara stabil hingga mendekati 0,98–0,99 tanpa jarak besar antara keduanya. Hal ini memperkuat observasi bahwa DeiT memiliki efisiensi pembelajaran dan kemampuan generalisasi yang lebih baik dibanding ViT pada dataset ini.

5.3 Analisis Confusion Matrix



Gambar 9: Confusion matrix model ViT pada fold terbaik



Gambar 10: Confusion matrix model DeiT pada fold terbaik

Gambar 9 dan Gambar 10 menampilkan confusion matrix untuk ViT dan DeiT pada fold terbaik (fold ke-4).

Pada ViT, diagonal utama hampir terisi penuh, menunjukkan tingkat prediksi yang tinggi untuk seluruh kelas. Beberapa pola kesalahan yang muncul antara lain:

- Kelas bakso: 72 prediksi benar, 2 sampel salah diklasifikasikan sebagai gado gado.
- Kelas gado gado: sebagian besar benar, dengan 1 sampel salah sebagai rendang.
- Kelas nasi goreng: 74 benar, 1 salah sebagai gado gado dan 1 sebagai soto ayam.
- Kelas soto ayam: 71 benar, 2 salah sebagai gado gado.

Kesalahan pada ViT tampak menyebar tipis ke beberapa kelas yang berdekatan secara visual, terutama antara gado gado, nasi goreng, dan soto ayam. Pola ini dapat diinterpretasikan sebagai tumpang tindih karakteristik visual antar kelas yang memang cukup mirip dari sisi warna, komposisi, dan keberadaan nasi atau kuah.

Untuk DeiT, pola diagonal pada confusion matrix lebih bersih dibandingkan ViT. Beberapa temuan utama adalah:

- Kelas bakso diklasifikasikan sepenuhnya benar (akurasi 100 persen pada fold ini).
- Kelas nasi goreng dan rendang menunjukkan akurasi sangat tinggi dengan hanya 1–2 kesalahan silang.
- Kesalahan relatif lebih terkonsentrasi pada kelas gado gado yang terkadang diprediksi sebagai nasi goreng, rendang, atau soto ayam.

Secara keseluruhan, DeiT menunjukkan ketegasan yang lebih tinggi dalam membedakan kelas dengan struktur visual yang kuat seperti bakso dan soto ayam, sedangkan tantangan terbesar tetap terdapat pada kelas gado gado yang memiliki variasi visual lebih besar dan sulit dibedakan secara kontras antar sampel.

5.4 Evaluasi Classification Report

	precision	recall	f1-score	support
bakso	0.986301	0.972973	0.979592	74.000000
gado_gado	0.935065	0.986301	0.960000	73.000000
nasi_goreng	1.000000	0.973684	0.986667	76.000000
rendang	0.986667	0.986667	0.986667	75.000000
soto_ayam	0.986111	0.972603	0.979310	73.000000
accuracy	0.978437	0.978437	0.978437	0.978437
macro avg	0.978829	0.978446	0.978447	371.000000
weighted avg	0.979062	0.978437	0.978561	371.000000

Gambar 11: Classification report model ViT

	precision	recall	f1-score	support
bakso	1.000000	1.000000	1.000000	74.000000
gado_gado	1.000000	0.945205	0.971831	73.000000
nasi_goreng	0.974026	0.986842	0.980392	76.000000
rendang	0.986667	0.986667	0.986667	75.000000
soto_ayam	0.960526	1.000000	0.979866	73.000000
accuracy	0.983827	0.983827	0.983827	0.983827
macro avg	0.984244	0.983743	0.983751	371.000000
weighted avg	0.984217	0.983827	0.983783	371.000000

Gambar 12: Classification report model DeiT

Gambar 11 dan Gambar 12 menyajikan classification report untuk ViT dan DeiT. Secara umum, seluruh kelas menunjukkan nilai precision, recall, dan F1-score yang tinggi pada kedua model. Namun terdapat beberapa perbedaan penting.

Pada ViT, sebagian besar kelas memperoleh F1-score pada rentang 0,96–0,98. Kinerja relatif lebih rendah terlihat pada kelas gado gado dibandingkan kelas lainnya. Sementara itu, DeiT memberikan F1-score mendekati 1,00 pada kelas dengan ciri visual yang kuat seperti bakso dan soto ayam, serta performa yang lebih konsisten pada seluruh kelas lainnya.

Perbedaan ini selaras dengan nilai macro F1 rata-rata, di mana DeiT mencapai 0,9747 sedangkan ViT berada pada 0,9598, sehingga mengindikasikan bahwa DeiT memiliki kemampuan klasifikasi yang sedikit lebih stabil di seluruh kelas.

5.5 Analisis Ringkasan Perbandingan Model

--- Ringkasan Perbandingan ViT vs DeiT (Rata-Rata 5 Fold) ---										
Model	Total Params	Trainable Params	Non-trainable Params	Model Size (MB)	Val Acc (mean 5-fold)	Macro Precision (mean)	Macro Recall (mean)	Macro F1 (mean)	Infer Avg Time ms/img (mean)	Infer Throughput img/s (mean)
0 ViT	85802501	85802501	0	327.31	0.9597	0.9621	0.9596	0.9598	10.2144	97.90
1 DeiT	21667589	21667589	0	82.66	0.9747	0.9756	0.9746	0.9747	2.8262	353.93

Gambar 13: Ringkasan perbandingan performa ViT dan DeiT

Gambar 13 menampilkan tabel ringkasan perbandingan performa kedua model. Terdapat tiga poin utama yang dapat disimpulkan dari hasil tersebut.

Parameter dan ukuran model: DeiT memiliki jumlah parameter sekitar 75 persen lebih sedikit dibandingkan ViT, sehingga ukuran model menjadi sekitar empat kali lebih kecil.

Performa klasifikasi: DeiT menunjukkan performa yang lebih baik pada seluruh metrik validasi, yaitu accuracy, macro precision, macro recall, dan macro F1, dengan selisih sekitar 1,5–1,7 poin persentase secara konsisten pada kelima fold.

Waktu inferensi: DeiT memiliki waktu inferensi yang jauh lebih cepat, yaitu rata-rata 2,83 ms per citra dibandingkan 10,21 ms per citra pada ViT. Hal ini setara dengan throughput 353,93 citra per detik pada DeiT, sedangkan ViT hanya mencapai 97,90 citra per detik.

Secara keseluruhan, hasil ini menunjukkan bahwa DeiT menghasilkan keseimbangan terbaik antara akurasi dan efisiensi runtime pada konfigurasi eksperimen yang digunakan.

5.6 Analisis Pribadi

Beberapa kesimpulan analitis yang diperoleh dari hasil eksperimen adalah sebagai berikut.

Pertama, model berkapasitas besar tidak selalu menghasilkan akurasi terbaik. Kapasitas ViT yang jauh lebih besar dibandingkan kebutuhan dataset menyebabkan munculnya overfitting ringan, sehingga peningkatan jumlah parameter tidak memberikan kenaikan akurasi yang signifikan.

Kedua, DeiT mencapai akurasi lebih tinggi karena efisiensi proses pembelajarannya. Pendekatan data-efficient training membuat DeiT mampu mempelajari pola visual penting secara optimal tanpa terlalu menyerap noise dari data.

Ketiga, efisiensi inferensi DeiT menjadikannya lebih sesuai untuk penggunaan nyata. Dengan latensi sekitar 3,6 kali lebih cepat dibandingkan ViT, DeiT lebih cocok untuk skenario real-time seperti kasir otomatis, kamera restoran, maupun aplikasi mobile.

Keempat, karakteristik dataset memengaruhi performa model pada tingkat per kelas. Kelas gado gado secara konsisten menjadi kelas yang paling sulit diklasifikasikan pada kedua model, diduga karena variasi tampilan yang paling tinggi dibandingkan kelas makanan lain.

Kelima, trade-off yang paling ideal bukan sekadar ukuran parameter terbesar, melainkan keseimbangan antara akurasi dan latency. Pada penelitian ini, DeiT memberikan trade-off terbaik di seluruh dimensi evaluasi.

6 Kesimpulan dan Saran

Penelitian ini membandingkan performa model Vision Transformer (ViT) dan Data-Efficient Image Transformer (DeiT) dalam klasifikasi citra makanan Indonesia menggunakan fine-tuning dan evaluasi 5-Fold Stratified Cross-Validation. Hasil eksperimen menunjukkan bahwa DeiT unggul secara konsisten pada seluruh aspek evaluasi. DeiT memperoleh rata-rata val accuracy sebesar 0,9747 dan macro F1 sebesar 0,9747, lebih tinggi dibandingkan ViT yang masing-masing mencapai 0,9597 dan 0,9598. Keunggulan tersebut dicapai meskipun ukuran DeiT jauh lebih kecil, yaitu sekitar 21,6 juta parameter (82,66 MB), sedangkan ViT berjumlah sekitar 85,8 juta parameter (327,31 MB). Dari sisi efisiensi, DeiT memiliki waktu inferensi rata-rata 2,83 ms per citra dengan throughput sekitar 354 citra per detik, jauh lebih cepat dibandingkan ViT yang membutuhkan 10,21 ms per citra dengan throughput sekitar 98 citra per detik. Analisis kurva belajar dan confusion matrix mengonfirmasi bahwa DeiT memiliki generalization gap yang lebih kecil serta stabilitas optimisasi yang lebih baik, sedangkan ViT menunjukkan kecenderungan overfitting ringan. Kedua model masih mengalami kesulitan pada kelas makanan dengan variasi visual tinggi, khususnya gado gado.

Berdasarkan temuan tersebut, DeiT dapat direkomendasikan sebagai pilihan yang paling sesuai untuk pengembangan sistem klasifikasi makanan Indonesia, terutama untuk aplikasi yang menuntut respons cepat seperti kasir otomatis atau aplikasi mobile. Arah penelitian berikutnya dapat mencakup: (1) perluasan dataset dan strategi augmentasi untuk mengatasi variasi tampilan kelas tertentu; (2) pengujian implementasi pada perangkat edge untuk menilai kelayakan penggunaan dunia nyata; (3) eksplorasi arsitektur lain seperti Swin, MobileViT, MAE, atau EfficientFormer; dan (4) integrasi pendekatan class-balanced loss atau attention visualization guna meningkatkan kinerja per kelas serta memperkuat interpretabilitas model.

7 Lampiran

Seluruh foto telah dimasukkan ke dalam laporan dan kode berada di link github.

References

- [1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- [2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” in *International Conference on Learning Representations (ICLR)*, 2021.
- [3] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, “Training data-efficient image transformers & distillation through attention,” in *International Conference on Machine Learning (ICML)*, 2021.
- [4] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *arXiv preprint arXiv:1503.02531*, 2015.
- [5] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby, “Big transfer (bit): General visual representation learning,” *arXiv preprint arXiv:1912.11370*, 2020.