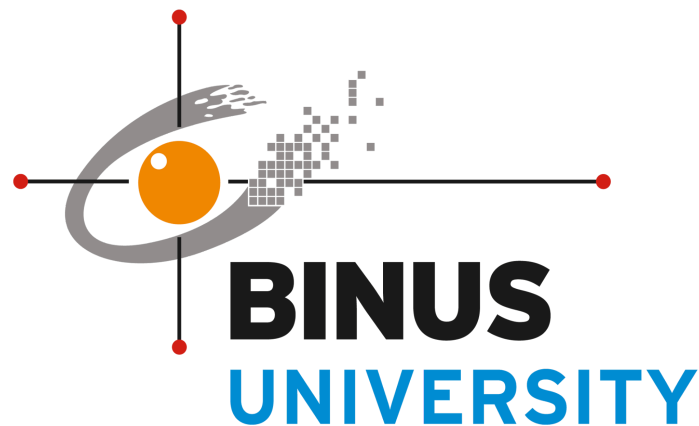


# LAPORAN PROJECT DEEP LEARNING



## IMPLEMENTASI DEEP LEARNING PADA KLASIFIKASI GAMBAR MOTIF BATIK: SIDOLUHUR, TAMBAL, DAN BETAWI

**Kelompok : 6**

**Nama dan NIM Anggota Kelompok :**

Alleluia Rehuellah Alefika Ndolu	2602056334	Data Science
Angela Hartono	2602059582	Data Science
Jessica Berliani	2602071285	Data Science

DEEP LEARNING  
KELAS: LA09  
TAHUN AJARAN: SEMESTER GENAP 2023/2024

# **1. Pendahuluan**

## **1.1. Permasalahan**

Di Indonesia, batik merupakan salah satu warisan budaya yang mengandung kekayaan motif yang sangat beragam dan unik. Setiap motif batik memiliki karakteristik visual tersendiri, termasuk pola, warna, dan bentuk yang membedakannya. Proses klasifikasi batik secara manual sering kali menantang dan memerlukan waktu yang lama, terutama ketika menghadapi koleksi batik yang besar dan beragam. Mengingat kompleksitas ini, teknologi kecerdasan buatan menjadi sangat penting untuk mendukung identifikasi dan klasifikasi motif batik secara otomatis. Penerapan teknologi ini tidak hanya memudahkan pengkategorian yang biasanya dilakukan oleh ahli batik, tetapi juga membantu dalam proses penyusunan katalog batik yang efektif, memungkinkan pengetahuan tentang berbagai jenis batik tersimpan secara sistematis dan dapat diakses dengan lebih mudah oleh berbagai pihak, termasuk peneliti, desainer, dan penggemar budaya.

## **1.2. Tugas yang Akan Dilakukan**

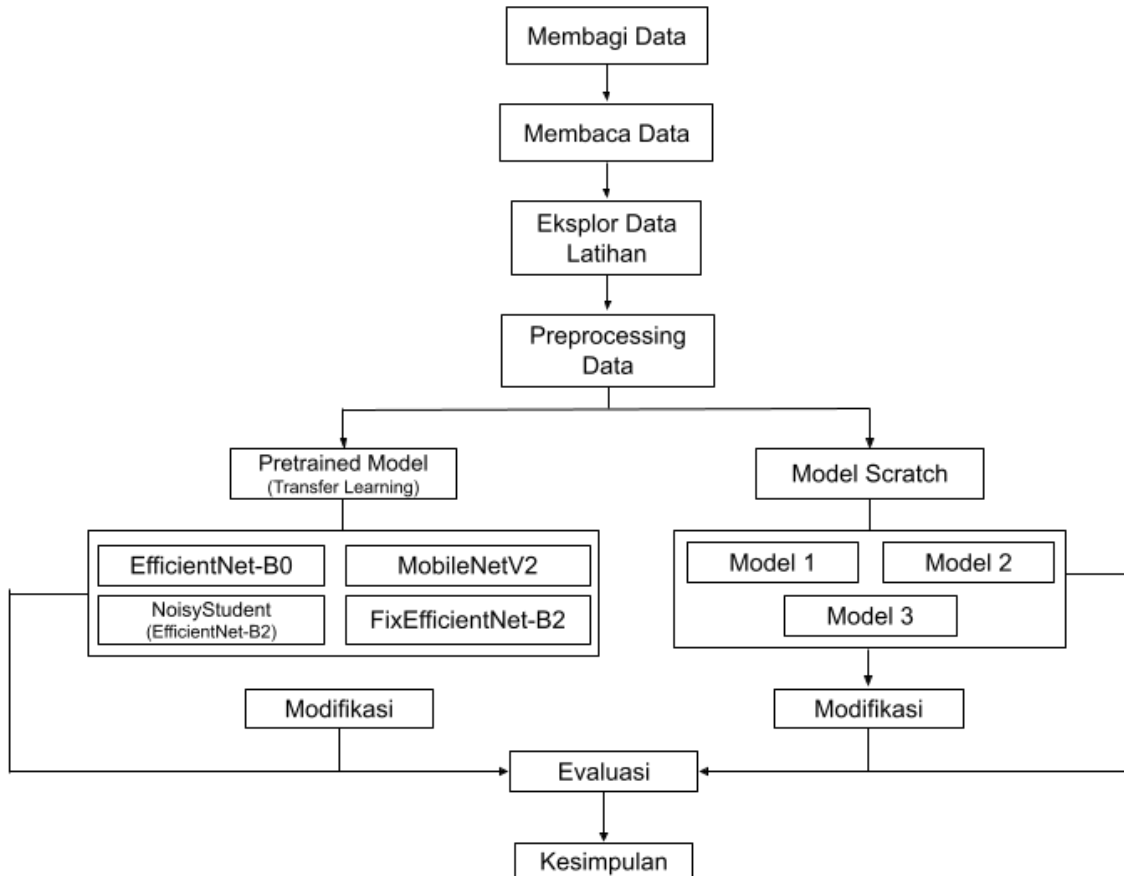
Tugas ini bertujuan untuk mengembangkan model pembelajaran mesin yang mampu melakukan klasifikasi motif batik. Fokus utamanya adalah pembuatan dua jenis model, pertama, model yang dirancang dari awal (scratch) dan kedua, model yang dibangun menggunakan teknik transfer learning dengan jumlah parameter kurang dari 10 juta. Setiap model ini dirancang untuk mengklasifikasikan gambar batik ke dalam salah satu dari tiga kategori motif batik yang telah ditentukan, yaitu Batik Sidoluhur, Batik Betawi, dan Batik Tambal. Proses ini melibatkan penggunaan gambar-gambar yang diinputkan ke dalam sistem dan dikategorikan sesuai dengan karakteristik motif yang telah diidentifikasi sebelumnya. Selanjutnya, kedua teknik tersebut akan dievaluasi untuk menentukan mana yang lebih efektif dalam menghasilkan kinerja klasifikasi yang akurat. Hasil evaluasi ini akan digunakan untuk menyimpulkan teknik mana yang paling efektif dalam mendukung proses klasifikasi batik, sehingga dapat memberikan wawasan baru dalam pengembangan teknologi pengenalan motif batik di Indonesia.

## **1.3. Analisis Data Sederhana**

Data terdiri dari gambar-gambar batik yang telah dikategorikan ke dalam beberapa kelas motif. Data yang digunakan bersumber dari website kaggle, dengan link sebagai berikut : <https://www.kaggle.com/dionisiusdh/indonesian-batik-motifs/data>. Jumlah data pada ketiga kelas yang terpilih berjumlah sama, yaitu ada 50 gambar. Gambar-gambar ini diproses dan digunakan untuk melatih model menggunakan pendekatan transfer learning, yang memungkinkan penggunaan model yang telah dilatih sebelumnya pada dataset besar untuk meningkatkan akurasi pada dataset yang lebih spesifik ini.

## 2. Metodologi

### 2.1. Metode Penyelesaian Masalah (Chart):



### 2.2. Proses Penyelesaian Masalah

Berikut adalah penjelasan dari setiap tahapan proses penyelesaian masalah yang kami lakukan sesuai gambar chart pada 2.1.

#### 1. Split Data

Kami melakukan pembagian data menjadi tiga bagian, dengan secara manual. Tiga bagian tersebut meliputi 80% data latihan (train data), 10% data validasi (validation data), dan 10% data pengujian (test data). Pembagian ini penting untuk memastikan bahwa model yang dikembangkan mampu generalisasi dengan baik pada data baru yang belum pernah dilihat selama proses pelatihan.

#### 2. Membaca Data

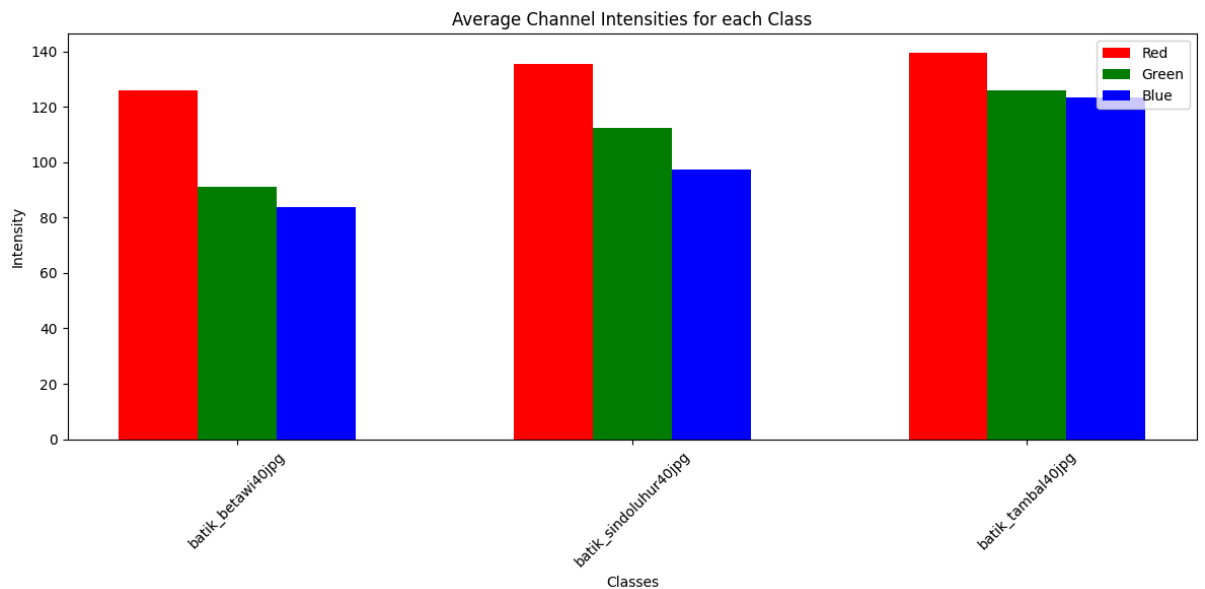
Proses dimulai dengan membaca dataset gambar batik yang telah dikumpulkan. Dataset ini disimpan dan diakses dari Google Drive, memastikan semua data terorganisir dan mudah diakses untuk proses selanjutnya.

### 3. Eksplorasi Data Latihan

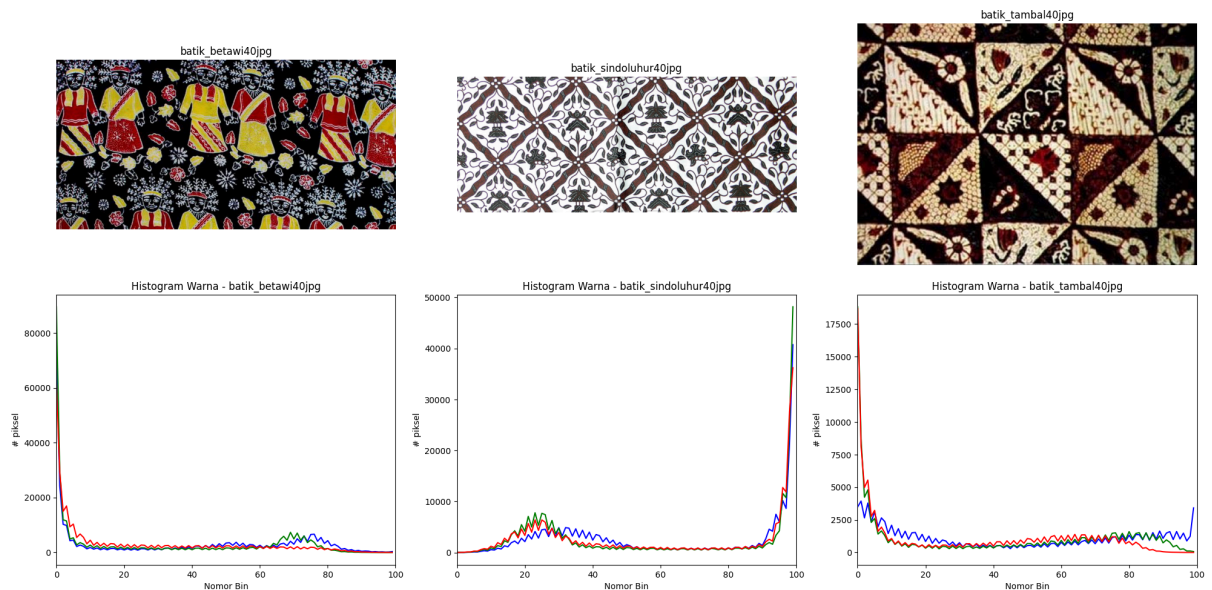
Kami melakukan eksplorasi pada data latihan untuk memahami variasi dalam gambar batik, seperti rata-rata intensitas RGB pada setiap kelas batik dan juga color histogram. Hal ini membantu kami dalam merencanakan langkah-langkah preprocessing yang sesuai.



Dimulai dengan melihat apakah data yang dibaca sudah benar, disini dimunculkan satu gambar batik untuk setiap kelas. Dari hasil ini dapat dilihat jika gambar memiliki ukuran yang beragam.



Rata - rata untuk intensitas warna RGB terlihat pada hasil diatas, warna merah merupakan warna yang paling dominan disetiap kelas, lalu di lanjut dengan warna hijau dan kemudian warna biru.



Histogram untuk batik\_betawi40.jpg:

- Saluran merah, hijau, dan biru memiliki jumlah piksel tinggi pada nilai intensitas rendah, menunjukkan bahwa gambar memiliki banyak area gelap atau hitam.
- Terdapat puncak yang mencolok pada saluran merah di sekitar nilai intensitas menengah, menunjukkan adanya warna merah yang menonjol dalam gambar.
- Saluran hijau dan biru juga menunjukkan pola serupa tetapi dengan intensitas yang lebih rendah dibandingkan dengan saluran merah.

Histogram untuk batik\_sindoluhur40.jpg:

- Saluran merah, hijau, dan biru menunjukkan jumlah piksel yang relatif seimbang, menunjukkan campuran warna.
- Terdapat puncak pada tingkat intensitas yang berbeda untuk setiap saluran, menunjukkan palet warna yang beragam dalam gambar.
- Saluran merah menunjukkan lebih banyak piksel pada intensitas yang lebih tinggi, menunjukkan area merah yang lebih terang.

Histogram untuk batik\_tambal40.jpg:

- Mirip dengan histogram pertama, terdapat jumlah piksel tinggi pada nilai intensitas rendah untuk semua saluran, menunjukkan area gelap.
- Saluran merah menunjukkan puncak pada nilai intensitas yang lebih tinggi, menunjukkan elemen merah yang menonjol.
- Saluran hijau dan biru memiliki intensitas piksel yang lebih rendah secara keseluruhan.

#### 4. Preprocessing Data

Data yang telah dieksplorasi kemudian diolah melalui langkah preprocessing, yang mencakup:

- **Melakukan augmentasi data** untuk meningkatkan variasi data latihan, seperti **random horizontal flip** dimana gambar akan dibalik secara horizontal secara acak. Hal ini membantu model untuk belajar dari gambar yang dilihat dari arah yang berbeda, meningkatkan kemampuan generalisasi model dengan meniru variasi posisi subjek yang mungkin terjadi dalam kumpulan data nyata.
- **Mengubah ukuran gambar menjadi 224x224 piksel.** Proses ini dilakukan untuk memastikan bahwa semua gambar memiliki dimensi yang seragam sebelum diproses oleh model pembelajaran mesin. Ukuran yang seragam penting untuk memastikan bahwa setiap input yang masuk ke dalam model memiliki struktur dan ukuran yang konsisten, yang memudahkan proses pelatihan model dan menghindari masalah komputasi yang mungkin terjadi karena variasi ukuran gambar.
- **Mengkonversi gambar menjadi tensor** merupakan proses penting dalam preprocessing khususnya dalam deep learning untuk proses pelatihan. Tensor adalah struktur data yang efektif untuk menyimpan gambar dalam bentuk yang dapat diproses oleh algoritma pembelajaran mesin, termasuk melaksanakan operasi matematis dan statistik yang diperlukan selama pelatihan sehingga dapat mempersiapkan data dengan cara yang memaksimalkan kemampuan belajar model, serta meningkatkan akurasi dan efektivitas model dalam mengklasifikasikan motif batik secara otomatis.
- **Normalisasi gambar** yang melibatkan penyesuaian skala piksel gambar agar nilai-nilainya berada dalam rentang yang lebih standar. Kami menggunakan `transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])` menyesuaikan setiap channel dari gambar RGB dengan cara mengurangi nilai mean dan membaginya dengan standar deviasi yang ditetapkan berdasarkan data ImageNet. Proses ini penting karena membantu dalam menyederhanakan model komputasi dengan menstandarisasi range nilai input, yang secara langsung mempengaruhi kecepatan konvergensi selama pelatihan dan membantu mengurangi kemungkinan overfitting. Selain itu, dengan mengurangi variasi internal gambar, model dapat lebih fokus pada pemahaman fitur penting daripada perbedaan kontras yang tidak penting, yang pada akhirnya meningkatkan akurasi dan efektivitas model dalam mengenali dan mengklasifikasikan motif batik secara akurat dan efisien.

Dengan menggunakan fungsi **'ImageFolder'**, kita memuat gambar dari direktori yang spesifik dan menerapkan transformasi preprocessing yang telah ditentukan sebelumnya untuk mempersiapkan data tersebut sesuai kebutuhan model. Selanjutnya, **'DataLoader'** digunakan untuk mengatur cara gambar tersebut disajikan ke model saat pelatihan dan evaluasi, seperti pengaturan jumlah gambar per batch dan apakah perlu pengacakan. Proses ini sangat penting untuk memastikan bahwa model dapat belajar dari data yang terstruktur dan konsisten, serta efisien dalam menangani volume data yang besar

## 5. Penggunaan, Modifikasi, dan Pelatihan Model

- **Penggunaan Model Pre-trained:**

Untuk teknik transfer learning, kami menggunakan arsitektur MobileNetV2 dan berbagai model EfficientNet, termasuk model seperti EfficientNet-B0, NoisyStudent

(EfficientNet-B2), dan FixEfficientNet-B2 yang telah dilatih sebelumnya untuk mengekstraksi fitur dari gambar batik. Model-model ini dipilih berdasarkan jumlah parameternya yang kurang dari 10 juta, serta tingkat akurasi yang tinggi, sesuai dengan hasil yang tercatat dalam benchmark ImageNet pada platform Papers with Code. Sumber ini menjamin efektivitas mereka dalam mengekstrak fitur dari gambar batik, sekaligus memastikan efisiensi komputasi yang diperlukan untuk aplikasi klasifikasi.

- **Penggunaan Model yang didesain dari scratch:**

Untuk teknik scratch, kami telah membuat model dari awal berdasarkan arsitektur CNN. Secara garis besar, terdapat 5 layer yang kami gunakan, yaitu:

1. Layer konvolusi (Conv): Berfungsi untuk mengekstraksi fitur-fitur dari gambar, seperti tepi, tekstur, dan pola yang lebih kompleks.
2. Pooling: Digunakan untuk mengurangi dimensi gambar, yang bertujuan untuk mengurangi kompleksitas komputasi dan mencegah overfitting.
3. Fungsi aktivasi ReLU: Ditambahkan untuk memberikan non-linearitas, memungkinkan model untuk mempelajari fitur-fitur yang lebih kompleks.
4. Layer flatten: Digunakan untuk meratakan output dari layer sebelumnya, mengubah data dari bentuk multi-dimensi menjadi satu dimensi.
5. Layer fully connected: Digunakan untuk menggabungkan semua fitur yang diekstraksi, menghasilkan prediksi akhir yang akurat untuk setiap kelas yang ditentukan.

Berikut ini merupakan deskripsi detail dari setiap model yang digunakan:

**Model 1:**

1. Conv1: Layer konvolusi pertama dengan 16 feature maps dan kernel size 3x3.
  - Tujuan: Mengekstraksi fitur dasar dari gambar input.
  - Padding: Same padding untuk menjaga ukuran gambar.
2. ReLU1: Fungsi aktivasi non-linear setelah Conv1.
  - Tujuan: Menambahkan non-linearitas, memungkinkan model untuk mempelajari fitur lebih kompleks.
3. MaxPool1: Layer max pooling dengan kernel size 2x2 dan stride 2.
  - Tujuan: Mengurangi dimensi spatial gambar, menjaga fitur penting, dan mengurangi risiko overfitting.
4. Conv2: Layer konvolusi kedua dengan 32 feature maps dan kernel size 3x3.
  - Tujuan: Mengekstraksi fitur lebih kompleks dari output layer sebelumnya.
  - Padding: Same padding untuk menjaga ukuran gambar.
5. ReLU2: Fungsi aktivasi non-linear setelah Conv2.
  - Tujuan: Menambahkan non-linearitas, memungkinkan model untuk mempelajari fitur lebih kompleks.
6. MaxPool2: Layer max pooling dengan kernel size 2x2 dan stride 2.

- Tujuan: Mengurangi dimensi spatial gambar, menjaga fitur penting, dan mengurangi risiko overfitting.
7. Flatten: Meratakan peta fitur menjadi vektor fitur 1D.
8. FC1: Layer fully connected pertama dengan 512 neuron.
- Tujuan: Menghubungkan hasil dari layer sebelumnya ke neuron-512 untuk mempelajari representasi yang lebih kompleks.
9. ReLU3: Fungsi aktivasi non-linear setelah FC1.
- Tujuan: Menambahkan non-linearitas, memungkinkan model untuk mempelajari fitur lebih kompleks.
10. FC2: Layer fully connected kedua dengan jumlah neuron sesuai dengan jumlah kelas (num\_classes).
- Tujuan: Menghasilkan output yang sesuai dengan jumlah kelas untuk tugas klasifikasi.

## **Model 2:**

1. Conv1: Layer konvolusi pertama dengan 64 feature maps dan kernel size 7x7.
- Tujuan: Mengekstraksi fitur dasar dari gambar input, menangkap informasi lebih luas dengan kernel besar.
2. Batch Normalization (bn1): Normalisasi batch setelah layer konvolusi pertama.
- Tujuan: Mencegah perubahan distribusi nilai yang ekstrem saat propagasi maju dan mundur, mempercepat konvergensi model.
3. ReLU (relu): Fungsi aktivasi ReLU setelah normalisasi batch.
- Tujuan: Menambah kapasitas non-linearitas ke dalam jaringan, mempercepat pembelajaran dengan mengaktifkan unit-unit yang tidak aktif dalam input.
4. MaxPool: Layer pooling dengan kernel size 3x3 dan stride 2.
- Tujuan: Mengurangi dimensi spatial gambar secara signifikan, mengurangi kompleksitas komputasi.
5. Layer1 - Layer4: Blok residual dengan output channels bervariasi (64, 128, 256, 512).
- Tujuan: Menggunakan shortcut connections untuk mengatasi masalah vanishing gradient, memungkinkan pelatihan jaringan lebih dalam.
  - Stride: Mengubah dimensi spatial sesuai kebutuhan untuk mengekstraksi fitur lebih kompleks.
6. AdaptiveAvgPool: Layer pooling adaptif yang mengubah ukuran output menjadi tetap 1x1.
- Tujuan: Meringkas seluruh fitur ke dalam satu vektor sebelum masuk ke layer fully connected.



7. Fully Connected (fc): Layer fully connected dengan output sesuai jumlah kelas.
  - Tujuan: Menghasilkan prediksi akhir untuk setiap kelas.

### **Model 3:**

1. Conv1 (self.conv1): Layer konvolusi pertama dengan 32 feature maps dan kernel size 3x3.
  - Tujuan: Mengekstraksi fitur dasar dari gambar input, seperti tepi dan tekstur.
  - Padding: Same padding digunakan agar ukuran gambar tetap.
2. ReLU1 (self.relu1): Fungsi aktivasi ReLU setelah Conv1.
  - Tujuan: Menambahkan non-linearitas ke dalam jaringan.
3. MaxPool1 (self.pool1): Layer pooling dengan kernel size 2x2 dan stride 2.
  - Tujuan: Mengurangi dimensi spatial gambar, mengurangi kompleksitas komputasi, dan menghindari overfitting.
4. Conv2 (self.conv2): Layer konvolusi kedua dengan 64 feature maps dan kernel size 3x3.
  - Tujuan: Mengekstraksi fitur lebih kompleks dari output layer sebelumnya.
  - Padding: Same padding digunakan agar ukuran gambar tetap.
5. ReLU2 (self.relu2): Fungsi aktivasi ReLU setelah Conv2.
  - Tujuan: Menambahkan non-linearitas ke dalam jaringan.
6. MaxPool2 (self.pool2): Layer pooling dengan kernel size 2x2 dan stride 2.
  - Tujuan: Mengurangi dimensi spatial gambar lebih lanjut, menjaga fitur penting, dan mengurangi risiko overfitting.
7. Flatten (self.flatten): Layer untuk mengubah tensor dari 4D ke 2D.
  - Tujuan: Mempersiapkan data untuk layer fully connected.
8. FC1 (self.fc1): Layer fully connected pertama dengan 128 neurons.
  - Tujuan: Menggabungkan fitur yang diekstraksi oleh layer konvolusi menjadi representasi yang lebih kompak.
9. ReLU3 (self.relu3): Fungsi aktivasi ReLU setelah FC1.
  - Tujuan: Menambahkan non-linearitas ke dalam jaringan.
10. FC2 (self.fc2): Layer fully connected kedua dengan output sesuai jumlah kelas.
  - Tujuan: Menghasilkan prediksi akhir untuk setiap kelas.

- **Modifikasi dan Tuning Model:**

Untuk memenuhi kebutuhan spesifik dari klasifikasi motif batik, kami melakukan modifikasi pada lapisan terakhir dari model-model tersebut. Adaptasi ini

memungkinkan kami untuk menyelaraskan arsitektur dengan jumlah kelas motif batik yang ada dalam dataset kami. Proses tuning parameter juga dilakukan untuk mengoptimalkan model agar mencapai performa yang optimal, dengan menyesuaikan berbagai parameter seperti laju pembelajaran dan hyperparameter lainnya. Terhusus untuk model yang didesain dari scratch, hyperparameter yang di tuning adalah jumlah epoch, dan penambahan dropout.

- **Pelatihan Model:**

Model-model ini dilatih menggunakan dataset batik yang telah kami siapkan. Proses pelatihan melibatkan model pre-trained untuk memahami dasar-dasar struktur model yang efektif untuk dataset ini. Pelatihan ini bertujuan untuk mengadaptasi bobot model pre-trained ke spesifikasi klasifikasi motif batik yang kami targetkan, memastikan bahwa model dapat mengenali dan mengklasifikasikan berbagai motif batik dengan akurat dan efektif.

## **6. Evaluasi Model**

Setiap model yang telah dilatih kemudian dievaluasi menggunakan data pengujian (test data) untuk melihat efektivitas model. Kami menggunakan metrik seperti akurasi, presisi, recall, dan f1-score untuk menilai performa setiap model. Evaluasi ini penting untuk menentukan model mana yang paling baik dalam mengklasifikasikan motif batik.

## **7. Kesimpulan**

Berdasarkan hasil evaluasi, model terbaik kemudian dipilih dari kedua teknik tersebut. Hasil-hasil ini disimpulkan dalam laporan akhir proyek, yang mencakup rekomendasi untuk penelitian lebih lanjut dan potensi peningkatan model.

## **3. Hasil dan Analisa**

### **3.1. Eksperimen dan Penyetelan Parameter**

- **Model Transfer Learning**

Untuk teknik transfer learning, kami menggunakan model MobileNetV2, EfficientNet-B0, NoisyStudent (EfficientNet-B2), dan FixEfficientNet-B2 dimodifikasi dengan penambahan dropout sebesar 0.5 dan normalisasi batch untuk meminimalisir overfitting. Model ini kemudian dilatih dengan berbagai konfigurasi parameter.

- **Model yang didesain dari scratch**

Untuk model yang didesain dari scratch, kami telah membuat model dari awal berdasarkan arsitektur CNN dimodifikasi dengan penambahan dropout sebesar 0.2 atau 0.5, dan penambahan jumlah epoch.

### **3.2. Analisis Hasil**

#### **A. Model Transfer Learning**

Dari eksperimen yang dilakukan dengan teknik transfer learning, kami menggunakan model MobileNetV2 dan berbagai model EfficientNet, baik dalam versi asli maupun versi yang telah dimodifikasi, berikut analisisnya:

### 1. Model EfficientNet-B0:

- **Versi Asli:** Model ini menunjukkan peningkatan yang konsisten selama pelatihan, dengan akurasi validasi maksimal 86.67% pada epoch ke-10. Namun, pada evaluasi, akurasi pengujian hanya mencapai 66.67%. Perbedaan ini menunjukkan bahwa meskipun model ini memiliki kemampuan untuk mempelajari data, masih diperlukan penyesuaian untuk memperbaiki kemampuan generalisasi.

```
Training the original EfficientNet-B0 model...
Epoch [1/10], Loss: 1.1291, Accuracy: 0.3250, Val Loss: 1.0903, Val Accuracy: 0.4000
Epoch [2/10], Loss: 0.9769, Accuracy: 0.6333, Val Loss: 0.9982, Val Accuracy: 0.6000
Epoch [3/10], Loss: 0.8470, Accuracy: 0.8250, Val Loss: 0.9216, Val Accuracy: 0.6667
Epoch [4/10], Loss: 0.7574, Accuracy: 0.8750, Val Loss: 0.8577, Val Accuracy: 0.6667
Epoch [5/10], Loss: 0.6793, Accuracy: 0.8500, Val Loss: 0.8066, Val Accuracy: 0.7333
Epoch [6/10], Loss: 0.6019, Accuracy: 0.8917, Val Loss: 0.7518, Val Accuracy: 0.6667
Epoch [7/10], Loss: 0.5586, Accuracy: 0.9250, Val Loss: 0.7034, Val Accuracy: 0.6667
Epoch [8/10], Loss: 0.5345, Accuracy: 0.9000, Val Loss: 0.6746, Val Accuracy: 0.8000
Epoch [9/10], Loss: 0.4829, Accuracy: 0.9250, Val Loss: 0.6484, Val Accuracy: 0.7333
Epoch [10/10], Loss: 0.4232, Accuracy: 0.9417, Val Loss: 0.5821, Val Accuracy: 0.8667
```

```
Evaluating the original EfficientNet-B0 model...
Accuracy: 66.67%
```

	precision	recall	f1-score	support
batik_betawi40jpg	0.75	0.60	0.67	5
batik_sindoluhur40jpg	0.71	1.00	0.83	5
batik_tambal40jpg	0.50	0.40	0.44	5
accuracy			0.67	15
macro avg	0.65	0.67	0.65	15
weighted avg	0.65	0.67	0.65	15

- **Versi Modifikasi:** Dengan penambahan dropout dan normalisasi batch, model mencapai akurasi pengujian yang lebih tinggi, yaitu 73.33%. Ini menandakan peningkatan dalam kemampuan generalisasi model.

```
Training the modified EfficientNet-B0 model with added dropout and batch normalization...
Epoch [1/10], Loss: 0.9632, Accuracy: 0.5500, Val Loss: 0.8415, Val Accuracy: 0.6667
Epoch [2/10], Loss: 0.2775, Accuracy: 0.9833, Val Loss: 0.6653, Val Accuracy: 0.7333
Epoch [3/10], Loss: 0.1245, Accuracy: 0.9750, Val Loss: 0.6400, Val Accuracy: 0.7333
Epoch [4/10], Loss: 0.0391, Accuracy: 1.0000, Val Loss: 0.6998, Val Accuracy: 0.7333
Epoch [5/10], Loss: 0.0265, Accuracy: 0.9917, Val Loss: 0.7251, Val Accuracy: 0.7333
Epoch [6/10], Loss: 0.0173, Accuracy: 1.0000, Val Loss: 0.6426, Val Accuracy: 0.8667
Epoch [7/10], Loss: 0.0084, Accuracy: 1.0000, Val Loss: 0.6267, Val Accuracy: 0.8667
Epoch [8/10], Loss: 0.0120, Accuracy: 1.0000, Val Loss: 0.6270, Val Accuracy: 0.8667
Epoch [9/10], Loss: 0.0089, Accuracy: 1.0000, Val Loss: 0.6540, Val Accuracy: 0.8000
Epoch [10/10], Loss: 0.0040, Accuracy: 1.0000, Val Loss: 0.4708, Val Accuracy: 0.8000
```

```
Evaluating the modified EfficientNet-B0 model...
Accuracy: 73.33%
```

	precision	recall	f1-score	support
batik_betawi5jpg	1.00	0.40	0.57	5
batik_sindoluhur5jpg	0.83	1.00	0.91	5
batik_tambal5jpg	0.57	0.80	0.67	5
accuracy			0.73	15
macro avg	0.80	0.73	0.72	15
weighted avg	0.80	0.73	0.72	15

### 2. Model MobileNetV2:

- **Versi Asli:** Model ini variatif dalam kinerjanya dengan puncak akurasi validasi 86.67% pada epoch ke-3, 7, 8, dan 9. Namun, pada evaluasi, akurasi

pengujian hanya mencapai menunjukkan bahwa model ini memiliki keterbatasan dalam generalisasi di lintas kategori batik, yang mencerminkan kebutuhan untuk penyesuaian lebih lanjut dalam arsitektur atau data training.

```
Training the MobileNetV2 model...
Epoch [1/10], Loss: 1.1270, Accuracy: 0.3917, Val Loss: 0.9859, Val Accuracy: 0.4667
Epoch [2/10], Loss: 0.9066, Accuracy: 0.5833, Val Loss: 0.8345, Val Accuracy: 0.6000
Epoch [3/10], Loss: 0.7855, Accuracy: 0.7250, Val Loss: 0.6944, Val Accuracy: 0.8667
Epoch [4/10], Loss: 0.7050, Accuracy: 0.7333, Val Loss: 0.6379, Val Accuracy: 0.8000
Epoch [5/10], Loss: 0.5730, Accuracy: 0.8417, Val Loss: 0.6038, Val Accuracy: 0.8000
Epoch [6/10], Loss: 0.5521, Accuracy: 0.8583, Val Loss: 0.5678, Val Accuracy: 0.8000
Epoch [7/10], Loss: 0.4885, Accuracy: 0.8333, Val Loss: 0.5196, Val Accuracy: 0.8667
Epoch [8/10], Loss: 0.4516, Accuracy: 0.8750, Val Loss: 0.4780, Val Accuracy: 0.8667
Epoch [9/10], Loss: 0.4301, Accuracy: 0.9083, Val Loss: 0.5031, Val Accuracy: 0.8667
Epoch [10/10], Loss: 0.3651, Accuracy: 0.9000, Val Loss: 0.5153, Val Accuracy: 0.8000
```

```
Evaluating the MobileNetV2 model...
Accuracy: 66.67%
```

	precision	recall	f1-score	support
batik_betawi5jpg	0.50	0.80	0.62	5
batik_sindoluhur5jpg	1.00	0.60	0.75	5
batik_tambal5jpg	0.75	0.60	0.67	5
accuracy			0.67	15
macro avg	0.75	0.67	0.68	15
weighted avg	0.75	0.67	0.68	15

- **Versi Modifikasi:** Dengan penambahan dropout dan normalisasi batch, model menunjukkan penurunan kinerjanya secara keseluruhan, dengan akurasi pengujian menurun drastis menjadi 33.33%. Hal ini menunjukkan bahwa modifikasi tersebut mungkin terlalu berat atau tidak cocok dengan jenis data ini.

```
Training the modified MobileNetV2 model with added dropout and batch normalization...
Epoch [1/10], Loss: 1.3040, Accuracy: 0.2750, Val Loss: 1.1370, Val Accuracy: 0.3333
Epoch [2/10], Loss: 1.2193, Accuracy: 0.3500, Val Loss: 1.1451, Val Accuracy: 0.1333
Epoch [3/10], Loss: 1.2462, Accuracy: 0.3083, Val Loss: 1.1397, Val Accuracy: 0.1333
Epoch [4/10], Loss: 1.2507, Accuracy: 0.3000, Val Loss: 1.1383, Val Accuracy: 0.2000
Epoch [5/10], Loss: 1.1988, Accuracy: 0.2833, Val Loss: 1.1276, Val Accuracy: 0.1333
Epoch [6/10], Loss: 1.2085, Accuracy: 0.3500, Val Loss: 1.1199, Val Accuracy: 0.3333
Epoch [7/10], Loss: 1.2141, Accuracy: 0.3083, Val Loss: 1.1440, Val Accuracy: 0.2667
Epoch [8/10], Loss: 1.1951, Accuracy: 0.3583, Val Loss: 1.1482, Val Accuracy: 0.2667
Epoch [9/10], Loss: 1.2303, Accuracy: 0.2917, Val Loss: 1.1205, Val Accuracy: 0.3333
Epoch [10/10], Loss: 1.2366, Accuracy: 0.3333, Val Loss: 1.1517, Val Accuracy: 0.2000
```

```
Evaluating the modified MobileNetV2 model...
Accuracy: 33.33%
```

	precision	recall	f1-score	support
batik_betawi5jpg	0.38	0.60	0.46	5
batik_sindoluhur5jpg	0.00	0.00	0.00	5
batik_tambal5jpg	0.40	0.40	0.40	5
accuracy			0.33	15
macro avg	0.26	0.33	0.29	15
weighted avg	0.26	0.33	0.29	15

### 3. Model NoisyStudent EfficientNet-B2:

- **Versi Asli:** Model ini menunjukkan akurasi validasi yang berfluktuasi, dengan nilai tertinggi 60.00% pada epoch ke-3 hingga 10. Pada evaluasi, akurasi pengujian hanya 40.00%, menunjukkan keterbatasan dalam generalisasi.

```

Training the NoisyStudent EfficientNet-B2 model...
Epoch [1/10], Loss: 1.0983, Accuracy: 0.3417, Val Loss: 1.0563, Val Accuracy: 0.4000
Epoch [2/10], Loss: 0.9598, Accuracy: 0.7167, Val Loss: 1.0086, Val Accuracy: 0.5333
Epoch [3/10], Loss: 0.8962, Accuracy: 0.7750, Val Loss: 0.9740, Val Accuracy: 0.6000
Epoch [4/10], Loss: 0.8049, Accuracy: 0.8583, Val Loss: 0.9394, Val Accuracy: 0.6000
Epoch [5/10], Loss: 0.7172, Accuracy: 0.9000, Val Loss: 0.9065, Val Accuracy: 0.6000
Epoch [6/10], Loss: 0.6468, Accuracy: 0.9333, Val Loss: 0.8647, Val Accuracy: 0.6000
Epoch [7/10], Loss: 0.6150, Accuracy: 0.9000, Val Loss: 0.8264, Val Accuracy: 0.6000
Epoch [8/10], Loss: 0.5705, Accuracy: 0.9333, Val Loss: 0.8167, Val Accuracy: 0.6000
Epoch [9/10], Loss: 0.5445, Accuracy: 0.9000, Val Loss: 0.7804, Val Accuracy: 0.6000
Epoch [10/10], Loss: 0.4772, Accuracy: 0.9333, Val Loss: 0.7328, Val Accuracy: 0.6000

```

```

Evaluating the NoisyStudent EfficientNet-B2 model...
Accuracy: 40.00%

```

	precision	recall	f1-score	support
batik_betawi5jpg	0.33	0.20	0.25	5
batik_sindoluhur5jpg	0.50	0.80	0.62	5
batik_tambal5jpg	0.25	0.20	0.22	5
accuracy			0.40	15
macro avg	0.36	0.40	0.36	15
weighted avg	0.36	0.40	0.36	15

- **Versi Modifikasi:** Dengan penambahan dropout dan normalisasi batch, model mengalami peningkatan signifikan dalam performa model, dengan akurasi pengujian yang meningkat menjadi 73.33%. Modifikasi ini berhasil mengatasi beberapa keterbatasan model awal, menandakan peningkatan dalam kemampuan generalisasi dan memberikan hasil yang lebih baik dalam presisi, recall, dan f1-score di lintas kategori.

```

Training the modified NoisyStudent EfficientNet-B2 model with added dropout and batch normalization...
Epoch [1/10], Loss: 1.0023, Accuracy: 0.4833, Val Loss: 0.8371, Val Accuracy: 0.6000
Epoch [2/10], Loss: 0.2621, Accuracy: 0.9667, Val Loss: 0.7473, Val Accuracy: 0.6667
Epoch [3/10], Loss: 0.0721, Accuracy: 0.9917, Val Loss: 0.7568, Val Accuracy: 0.6667
Epoch [4/10], Loss: 0.0468, Accuracy: 0.9833, Val Loss: 0.8442, Val Accuracy: 0.6667
Epoch [5/10], Loss: 0.0329, Accuracy: 0.9917, Val Loss: 0.9100, Val Accuracy: 0.7333
Epoch [6/10], Loss: 0.0305, Accuracy: 0.9917, Val Loss: 0.7106, Val Accuracy: 0.8000
Epoch [7/10], Loss: 0.0078, Accuracy: 1.0000, Val Loss: 0.5381, Val Accuracy: 0.8667
Epoch [8/10], Loss: 0.0060, Accuracy: 1.0000, Val Loss: 0.4193, Val Accuracy: 0.8667
Epoch [9/10], Loss: 0.0050, Accuracy: 1.0000, Val Loss: 0.3682, Val Accuracy: 0.9333
Epoch [10/10], Loss: 0.0033, Accuracy: 1.0000, Val Loss: 0.2568, Val Accuracy: 0.9333

```

```

Evaluating the modified NoisyStudent EfficientNet-B2 model...
Accuracy: 73.33%

```

	precision	recall	f1-score	support
batik_betawi5jpg	1.00	0.40	0.57	5
batik_sindoluhur5jpg	0.71	1.00	0.83	5
batik_tambal5jpg	0.67	0.80	0.73	5
accuracy			0.73	15
macro avg	0.79	0.73	0.71	15
weighted avg	0.79	0.73	0.71	15

#### 4. Model FixEfficientNet-B2:

- **Versi Asli:** Model ini menunjukkan kinerja yang sangat baik selama pelatihan dengan akurasi validasi tertinggi 100% pada epoch ke-8. Namun, pada evaluasi, akurasi pengujian adalah 86.67%, yang menunjukkan sedikit perbedaan antara pelatihan dan pengujian yang mungkin mengindikasikan overfitting.

```

Training the FixEfficientNet-B2 model...
Epoch 1: Train Loss: 0.9341, Train Acc: 55.00%, Val Loss: 0.7134, Val Acc: 66.67%
Epoch 2: Train Loss: 0.2613, Train Acc: 99.17%, Val Loss: 0.4569, Val Acc: 73.33%
Epoch 3: Train Loss: 0.0626, Train Acc: 100.00%, Val Loss: 0.2107, Val Acc: 93.33%
Epoch 4: Train Loss: 0.0133, Train Acc: 100.00%, Val Loss: 0.1716, Val Acc: 93.33%
Epoch 5: Train Loss: 0.0172, Train Acc: 100.00%, Val Loss: 0.1794, Val Acc: 86.67%
Epoch 6: Train Loss: 0.0046, Train Acc: 100.00%, Val Loss: 0.3667, Val Acc: 86.67%
Epoch 7: Train Loss: 0.0408, Train Acc: 99.17%, Val Loss: 0.0771, Val Acc: 93.33%
Epoch 8: Train Loss: 0.0351, Train Acc: 99.17%, Val Loss: 0.0518, Val Acc: 100.00%
Epoch 9: Train Loss: 0.0056, Train Acc: 100.00%, Val Loss: 0.1250, Val Acc: 93.33%
Epoch 10: Train Loss: 0.0194, Train Acc: 99.17%, Val Loss: 0.1260, Val Acc: 93.33%

```

Evaluating the FixEfficientNet-B2 model...

Accuracy: 86.67%

	precision	recall	f1-score	support
batik_betawi5jpg	1.00	0.60	0.75	5
batik_sindoluhur5jpg	1.00	1.00	1.00	5
batik_tambal5jpg	0.71	1.00	0.83	5
accuracy			0.87	15
macro avg	0.90	0.87	0.86	15
weighted avg	0.90	0.87	0.86	15

- **Versi Modifikasi:** Dengan penambahan dropout dan normalisasi batch, model mengalami penurunan performa dalam pengujian menjadi 73.33%. Meskipun bertujuan meningkatkan generalisasi, penurunan ini menunjukkan bahwa modifikasi tersebut mengurangi kemampuan klasifikasi model secara keseluruhan.

Training the modified FixEfficientNet-B2 model with added dropout and batch normalization...

```

Epoch 1: Train Loss: 0.9547, Train Acc: 58.33%, Val Loss: 0.7449, Val Acc: 80.00%
Epoch 2: Train Loss: 0.2684, Train Acc: 94.17%, Val Loss: 0.5648, Val Acc: 86.67%
Epoch 3: Train Loss: 0.1123, Train Acc: 98.33%, Val Loss: 0.4782, Val Acc: 86.67%
Epoch 4: Train Loss: 0.0291, Train Acc: 100.00%, Val Loss: 0.5697, Val Acc: 86.67%
Epoch 5: Train Loss: 0.0181, Train Acc: 100.00%, Val Loss: 0.7298, Val Acc: 86.67%
Epoch 6: Train Loss: 0.0226, Train Acc: 100.00%, Val Loss: 0.8586, Val Acc: 80.00%
Epoch 7: Train Loss: 0.0189, Train Acc: 99.17%, Val Loss: 0.9868, Val Acc: 73.33%
Epoch 8: Train Loss: 0.0040, Train Acc: 100.00%, Val Loss: 1.1150, Val Acc: 73.33%
Epoch 9: Train Loss: 0.0042, Train Acc: 100.00%, Val Loss: 1.2916, Val Acc: 73.33%
Epoch 10: Train Loss: 0.0074, Train Acc: 100.00%, Val Loss: 1.1260, Val Acc: 73.33%

```

Evaluating the modified FixEfficientNet-B2 model...

Accuracy: 73.33%

	precision	recall	f1-score	support
batik_betawi5jpg	0.67	0.80	0.73	5
batik_sindoluhur5jpg	0.83	1.00	0.91	5
batik_tambal5jpg	0.67	0.40	0.50	5
accuracy			0.73	15
macro avg	0.72	0.73	0.71	15
weighted avg	0.72	0.73	0.71	15

## B. Model yang didesain dari scratch

Dari eksperimen yang dilakukan dengan teknik scratch, kami telah membuat model dari awal berdasarkan arsitektur CNN, ResNet, dan SimpleNet. baik dalam versi asli maupun versi yang telah dimodifikasi, berikut analisisnya:

### 1. Model 1

- **Versi Asli:** Dari 10 epoch yang diamati, model mengalami peningkatan akurasi dari sekitar 45% ke sekitar 84% pada data pelatihan. Namun, akurasi pada data validasi hanya berkisar 33% hingga 53%, sementara akurasi pada data uji sekitar 67%. Loss pada data pelatihan terus menurun, sedangkan loss pada data validasi mengalami fluktuasi dengan nilai yang lebih tinggi pada beberapa epoch awal. Ini menunjukkan adanya overfitting pada model terhadap data pelatihan dan kesulitan dalam generalisasi pada data baru.

```
Epoch 1/10
Train Loss: 10.4875 Acc: 0.4500
Val Loss: 11.2154 Acc: 0.3333
Epoch 2/10
Train Loss: 5.5285 Acc: 0.3500
Val Loss: 2.5436 Acc: 0.3333
Epoch 3/10
Train Loss: 1.7270 Acc: 0.5250
Val Loss: 2.2121 Acc: 0.2667
Epoch 4/10
Train Loss: 1.0955 Acc: 0.6250
Val Loss: 1.4012 Acc: 0.3333
Epoch 5/10
Train Loss: 0.7796 Acc: 0.7083
Val Loss: 1.0316 Acc: 0.4000
Epoch 6/10
Train Loss: 0.7636 Acc: 0.7583
Val Loss: 1.0589 Acc: 0.4000
Epoch 7/10
Train Loss: 0.7336 Acc: 0.7250
Val Loss: 1.1155 Acc: 0.3333
Epoch 8/10
Train Loss: 0.6185 Acc: 0.7667
Val Loss: 1.0191 Acc: 0.5333
Epoch 9/10
Train Loss: 0.4703 Acc: 0.8333
Val Loss: 1.2009 Acc: 0.4000
Epoch 10/10
Train Loss: 0.3709 Acc: 0.8417
Val Loss: 1.3865 Acc: 0.4000
Test Loss: 0.8804 Acc: 0.6667
```

- **Versi Modifikasi dengan menambah jumlah epoch:** Selama 20 epoch, akurasi pelatihan model meningkat secara signifikan dari 35% menjadi 100%, menunjukkan bahwa model mampu mempelajari data pelatihan dengan sangat baik. Namun, akurasi validasi tetap tidak konsisten, berkisar antara 33% hingga 53%, menunjukkan bahwa model mengalami overfitting. Ini juga tercermin pada loss validasi yang fluktuatif, dengan nilai yang awalnya menurun tetapi kemudian meningkat hingga mencapai 1.7105 pada epoch terakhir.

Epoch 1/20	Epoch 11/20
Train Loss: 19.0842 Acc: 0.3500	Train Loss: 0.5137 Acc: 0.8917
Val Loss: 22.9644 Acc: 0.3333	Val Loss: 1.0740 Acc: 0.4667
Epoch 2/20	Epoch 12/20
Train Loss: 11.5040 Acc: 0.3833	Train Loss: 0.4200 Acc: 0.9167
Val Loss: 2.8146 Acc: 0.4000	Val Loss: 1.1194 Acc: 0.4667
Epoch 3/20	Epoch 13/20
Train Loss: 2.3012 Acc: 0.4250	Train Loss: 0.3391 Acc: 0.9417
Val Loss: 1.9567 Acc: 0.3333	Val Loss: 1.1898 Acc: 0.4667
Epoch 4/20	Epoch 14/20
Train Loss: 1.3060 Acc: 0.4250	Train Loss: 0.2529 Acc: 0.9500
Val Loss: 1.1196 Acc: 0.3333	Val Loss: 1.3114 Acc: 0.4667
Epoch 5/20	Epoch 15/20
Train Loss: 0.8889 Acc: 0.5667	Train Loss: 0.1706 Acc: 0.9583
Val Loss: 1.1994 Acc: 0.4000	Val Loss: 1.3991 Acc: 0.4000
Epoch 6/20	Epoch 16/20
Train Loss: 0.8846 Acc: 0.6250	Train Loss: 0.1072 Acc: 0.9917
Val Loss: 1.1357 Acc: 0.5333	Val Loss: 1.4181 Acc: 0.3333
Epoch 7/20	Epoch 17/20
Train Loss: 0.8265 Acc: 0.6500	Train Loss: 0.0612 Acc: 0.9917
Val Loss: 1.0581 Acc: 0.5333	Val Loss: 1.5509 Acc: 0.3333
Epoch 8/20	Epoch 18/20
Train Loss: 0.7686 Acc: 0.7167	Train Loss: 0.0298 Acc: 1.0000
Val Loss: 1.0092 Acc: 0.4667	Val Loss: 1.5773 Acc: 0.4000
Epoch 9/20	Epoch 19/20
Train Loss: 0.7026 Acc: 0.7917	Train Loss: 0.0225 Acc: 1.0000
Val Loss: 0.9786 Acc: 0.5333	Val Loss: 1.5663 Acc: 0.4000
Epoch 10/20	Epoch 20/20
Train Loss: 0.6087 Acc: 0.8333	Train Loss: 0.0103 Acc: 1.0000
Val Loss: 1.0107 Acc: 0.4000	Val Loss: 1.7105 Acc: 0.4000
	Test Loss: 0.8598 Acc: 0.6000

- Versi Modifikasi dengan menambah dropout sebesar 0.2:** Dengan menambahkan dropout 0.2, model menunjukkan peningkatan yang lebih lambat dalam akurasi pelatihan, mulai dari 37.5% pada epoch pertama hingga mencapai 100% pada epoch terakhir. Namun, performa pada data validasi tetap tidak stabil, dengan akurasi yang berkisar antara 20% hingga 47%, yang menunjukkan bahwa masalah overfitting masih ada meskipun sudah diterapkan dropout. Hal ini juga terlihat dari nilai loss validasi yang fluktuatif dan bahkan meningkat pada beberapa epoch. Secara keseluruhan, penambahan dropout sedikit membantu dalam mengurangi overfitting, namun performa validasi dan uji masih jauh dari memuaskan.



```
Epoch 1/10
Train Loss: 9.1620 Acc: 0.3750
Val Loss: 5.6019 Acc: 0.3333
Epoch 2/10
Train Loss: 3.0174 Acc: 0.4083
Val Loss: 2.1868 Acc: 0.3333
Epoch 3/10
Train Loss: 1.2811 Acc: 0.4750
Val Loss: 1.3973 Acc: 0.4000
Epoch 4/10
Train Loss: 0.9126 Acc: 0.6250
Val Loss: 1.2580 Acc: 0.2667
Epoch 5/10
Train Loss: 0.7198 Acc: 0.7417
Val Loss: 1.1942 Acc: 0.2000
Epoch 6/10
Train Loss: 0.5463 Acc: 0.8333
Val Loss: 1.2031 Acc: 0.4667
Epoch 7/10
Train Loss: 0.2951 Acc: 0.9417
Val Loss: 1.4066 Acc: 0.2667
Epoch 8/10
Train Loss: 0.1964 Acc: 0.9667
Val Loss: 1.7034 Acc: 0.3333
Epoch 9/10
Train Loss: 0.2108 Acc: 0.9083
Val Loss: 1.3745 Acc: 0.4667
Epoch 10/10
Train Loss: 0.0706 Acc: 1.0000
Val Loss: 1.2116 Acc: 0.4000
Test Loss: 1.1681 Acc: 0.4667
```

- **Versi Modifikasi dengan menambah dropout sebesar 0.5:** Dengan menambahkan dropout 0.5, model menunjukkan peningkatan akurasi pelatihan dari 34.17% pada epoch pertama hingga 85.83% pada epoch kesepuluh. Namun, akurasi validasi tetap tidak stabil dan berkisar antara 20% hingga 46.67%, menunjukkan bahwa masalah overfitting masih ada. Meskipun dropout 0.5 membantu dalam mengurangi overfitting, peningkatan pada data validasi tidak signifikan, dengan akurasi yang masih rendah dan fluktuatif. Secara keseluruhan, penambahan dropout 0.5 sedikit membantu dalam mengurangi overfitting, namun performa validasi dan uji tetap belum memuaskan.

```

Epoch 1/10
Train Loss: 12.8002 Acc: 0.3417
Val Loss: 11.1812 Acc: 0.3333
Epoch 2/10
Train Loss: 5.6927 Acc: 0.4083
Val Loss: 2.4703 Acc: 0.3333
Epoch 3/10
Train Loss: 1.6199 Acc: 0.4667
Val Loss: 1.6763 Acc: 0.3333
Epoch 4/10
Train Loss: 1.0858 Acc: 0.4500
Val Loss: 1.1459 Acc: 0.2000
Epoch 5/10
Train Loss: 0.9412 Acc: 0.6000
Val Loss: 1.0442 Acc: 0.4667
Epoch 6/10
Train Loss: 0.9161 Acc: 0.6417
Val Loss: 0.9898 Acc: 0.4000
Epoch 7/10
Train Loss: 0.8232 Acc: 0.6917
Val Loss: 0.9488 Acc: 0.4000
Epoch 8/10
Train Loss: 0.7335 Acc: 0.6667
Val Loss: 0.9602 Acc: 0.4000
Epoch 9/10
Train Loss: 0.6272 Acc: 0.8083
Val Loss: 0.9898 Acc: 0.4667
Epoch 10/10
Train Loss: 0.4992 Acc: 0.8583
Val Loss: 1.0773 Acc: 0.3333
Test Loss: 1.2563 Acc: 0.4000

```

- Versi Modifikasi dengan menambah dropout sebesar 0.2 dan menambah jumlah epoch:** Dengan penambahan dropout sebesar 0.2 dan peningkatan jumlah epoch menjadi 20, terlihat bahwa model mengalami peningkatan akurasi pada data pelatihan dari 41.67% menjadi 100%. Namun, performa pada data validasi masih fluktuatif dengan akurasi berkisar antara 13.33% hingga 53.33%, menunjukkan bahwa model masih mengalami overfitting. Pada data uji, model mencapai akurasi 80% dengan loss sebesar 1.2280. Meskipun dropout 0.2 membantu mengurangi overfitting dibandingkan dengan sebelumnya, model masih memiliki kesulitan dalam menggeneralisasi dengan baik pada data baru.

Epoch 1/20	Epoch 11/20
Train Loss: 11.3848 Acc: 0.4167	Train Loss: 0.1515 Acc: 0.9833
Val Loss: 10.4569 Acc: 0.4000	Val Loss: 1.3842 Acc: 0.3333
Epoch 2/20	Epoch 12/20
Train Loss: 6.1116 Acc: 0.4667	Train Loss: 0.0930 Acc: 1.0000
Val Loss: 4.4815 Acc: 0.2667	Val Loss: 1.3983 Acc: 0.4000
Epoch 3/20	Epoch 13/20
Train Loss: 2.3889 Acc: 0.4917	Train Loss: 0.0588 Acc: 1.0000
Val Loss: 1.7794 Acc: 0.2667	Val Loss: 1.4526 Acc: 0.4000
Epoch 4/20	Epoch 14/20
Train Loss: 0.8593 Acc: 0.6667	Train Loss: 0.0454 Acc: 1.0000
Val Loss: 1.4006 Acc: 0.1333	Val Loss: 1.4006 Acc: 0.4667
Epoch 5/20	Epoch 15/20
Train Loss: 0.6910 Acc: 0.6583	Train Loss: 0.0227 Acc: 1.0000
Val Loss: 1.1977 Acc: 0.2000	Val Loss: 1.6381 Acc: 0.4000
Epoch 6/20	Epoch 16/20
Train Loss: 0.5441 Acc: 0.8250	Train Loss: 0.0285 Acc: 0.9917
Val Loss: 1.1811 Acc: 0.2667	Val Loss: 1.4963 Acc: 0.2667
Epoch 7/20	Epoch 17/20
Train Loss: 0.4712 Acc: 0.8750	Train Loss: 0.0184 Acc: 1.0000
Val Loss: 1.1419 Acc: 0.2667	Val Loss: 1.3345 Acc: 0.5333
Epoch 8/20	Epoch 18/20
Train Loss: 0.4011 Acc: 0.9083	Train Loss: 0.0101 Acc: 1.0000
Val Loss: 1.1714 Acc: 0.2000	Val Loss: 1.4096 Acc: 0.4667
Epoch 9/20	Epoch 19/20
Train Loss: 0.3063 Acc: 0.9500	Train Loss: 0.0053 Acc: 1.0000
Val Loss: 1.2481 Acc: 0.3333	Val Loss: 1.6237 Acc: 0.4667
Epoch 10/20	Epoch 20/20
Train Loss: 0.2093 Acc: 0.9750	Train Loss: 0.0043 Acc: 1.0000
Val Loss: 1.3710 Acc: 0.2667	Val Loss: 1.5613 Acc: 0.4667
	Test Loss: 1.2280 Acc: 0.8000

- Versi Modifikasi dengan menambah dropout sebesar 0.5 dan menambah jumlah epoch:** Dengan penambahan dropout sebesar 0.5 dan jumlah epoch menjadi 20, hasil menunjukkan bahwa model mencapai akurasi 99.17% pada data pelatihan, namun akurasi pada data validasi tetap fluktuatif dan rendah, berkisar antara 13.33% hingga 33.33%. Akurasi pada data uji adalah 53.33% dengan loss sebesar 1.5952. Meskipun dropout 0.5 membantu dalam mengurangi overfitting dibandingkan dengan tanpa dropout, model masih menunjukkan kesulitan dalam menggeneralisasi dengan baik pada data validasi dan uji. Nilai loss pada data pelatihan terus menurun, tetapi loss pada data validasi meningkat secara signifikan setelah beberapa epoch, menunjukkan bahwa model mungkin terlalu kompleks untuk dataset ini, meskipun dengan regularisasi yang lebih tinggi melalui dropout.

Epoch 1/20	Epoch 11/20
Train Loss: 16.5339 Acc: 0.2833	Train Loss: 0.3771 Acc: 0.8917
Val Loss: 15.0029 Acc: 0.3333	Val Loss: 1.4153 Acc: 0.2667
Epoch 2/20	Epoch 12/20
Train Loss: 5.8422 Acc: 0.5167	Train Loss: 0.3312 Acc: 0.9000
Val Loss: 3.8360 Acc: 0.2667	Val Loss: 1.6625 Acc: 0.2000
Epoch 3/20	Epoch 13/20
Train Loss: 2.4149 Acc: 0.4667	Train Loss: 0.2737 Acc: 0.9083
Val Loss: 2.1027 Acc: 0.2000	Val Loss: 1.8286 Acc: 0.2667
Epoch 4/20	Epoch 14/20
Train Loss: 1.1418 Acc: 0.5500	Train Loss: 0.2403 Acc: 0.9500
Val Loss: 1.4418 Acc: 0.2000	Val Loss: 1.7452 Acc: 0.2667
Epoch 5/20	Epoch 15/20
Train Loss: 0.8740 Acc: 0.5500	Train Loss: 0.1774 Acc: 0.9583
Val Loss: 1.2488 Acc: 0.2000	Val Loss: 1.9269 Acc: 0.2667
Epoch 6/20	Epoch 16/20
Train Loss: 0.8070 Acc: 0.6750	Train Loss: 0.1549 Acc: 0.9417
Val Loss: 1.1149 Acc: 0.4000	Val Loss: 1.9676 Acc: 0.2000
Epoch 7/20	Epoch 17/20
Train Loss: 0.7194 Acc: 0.7250	Train Loss: 0.1755 Acc: 0.9667
Val Loss: 1.1718 Acc: 0.2000	Val Loss: 1.8964 Acc: 0.2667
Epoch 8/20	Epoch 18/20
Train Loss: 0.6220 Acc: 0.7667	Train Loss: 0.1156 Acc: 0.9667
Val Loss: 1.3565 Acc: 0.2667	Val Loss: 1.8750 Acc: 0.2667
Epoch 9/20	Epoch 19/20
Train Loss: 0.5795 Acc: 0.8250	Train Loss: 0.1021 Acc: 0.9833
Val Loss: 1.3921 Acc: 0.1333	Val Loss: 2.1161 Acc: 0.2667
Epoch 10/20	Epoch 20/20
Train Loss: 0.4334 Acc: 0.8583	Train Loss: 0.0693 Acc: 0.9917
Val Loss: 1.4053 Acc: 0.2000	Val Loss: 2.2314 Acc: 0.3333
	Test Loss: 1.5952 Acc: 0.5333

## 2. Model 2

- Versi Asli:** Terlihat bahwa model mengalami peningkatan akurasi pada data pelatihan dari 29.17% menjadi 69.17%. Namun, performa pada data validasi masih fluktuatif dengan akurasi berkisar antara 33.33% hingga 46.67%, menunjukkan bahwa model masih mengalami overfitting. Pada data uji, model mencapai akurasi 20% dengan loss sebesar 2.0919. Loss pada data pelatihan menunjukkan penurunan yang konsisten, tetapi loss pada data validasi tidak menunjukkan pola penurunan yang stabil, tetap berada di sekitar 1.1972 pada akhir epoch.

```
Epoch 1/10
Train Loss: 2.3694 Acc: 0.2917
Val Loss: 56.0374 Acc: 0.3333
Epoch 2/10
Train Loss: 1.2079 Acc: 0.3250
Val Loss: 211.6748 Acc: 0.3333
Epoch 3/10
Train Loss: 1.1298 Acc: 0.4333
Val Loss: 91.1631 Acc: 0.3333
Epoch 4/10
Train Loss: 0.9525 Acc: 0.5750
Val Loss: 23.6431 Acc: 0.4000
Epoch 5/10
Train Loss: 0.8438 Acc: 0.6083
Val Loss: 6.7712 Acc: 0.4000
Epoch 6/10
Train Loss: 0.7736 Acc: 0.6667
Val Loss: 2.0510 Acc: 0.4667
Epoch 7/10
Train Loss: 0.7771 Acc: 0.6500
Val Loss: 2.3911 Acc: 0.4000
Epoch 8/10
Train Loss: 0.7647 Acc: 0.6750
Val Loss: 2.5269 Acc: 0.4667
Epoch 9/10
Train Loss: 0.7751 Acc: 0.6500
Val Loss: 1.3600 Acc: 0.4667
Epoch 10/10
Train Loss: 0.6824 Acc: 0.6917
Val Loss: 1.1972 Acc: 0.4000
Test Loss: 2.0919 Acc: 0.2000
```

- **Versi Modifikasi dengan menambah jumlah epoch:** Pada penambahan epoch menjadi 20, terlihat bahwa model mengalami peningkatan dalam akurasi pada data pelatihan dari 33.33% menjadi 82.50%, menunjukkan adanya peningkatan performa yang signifikan. Namun, performa pada data validasi masih fluktuatif dengan akurasi yang berubah-ubah antara 13.33% hingga 66.67%. Hal ini menunjukkan bahwa model masih belum sepenuhnya mampu menggeneralisasi dengan baik pada data baru, terutama mengingat fluktuasi yang cukup besar pada akurasi data validasi. Pada data uji, model mencapai akurasi sebesar 66.67% dengan loss sebesar 1.3104.

Epoch 1/20	Epoch 11/20
Train Loss: 1.9506 Acc: 0.3333	Train Loss: 0.5925 Acc: 0.7333
Val Loss: 1.8377 Acc: 0.4000	Val Loss: 1.0396 Acc: 0.6000
Epoch 2/20	Epoch 12/20
Train Loss: 1.7648 Acc: 0.4167	Train Loss: 0.4762 Acc: 0.8083
Val Loss: 1.4380 Acc: 0.1333	Val Loss: 1.2972 Acc: 0.3333
Epoch 3/20	Epoch 13/20
Train Loss: 1.1450 Acc: 0.4167	Train Loss: 0.4933 Acc: 0.7917
Val Loss: 2.2721 Acc: 0.2000	Val Loss: 1.2535 Acc: 0.5333
Epoch 4/20	Epoch 14/20
Train Loss: 0.9333 Acc: 0.5750	Train Loss: 0.5326 Acc: 0.8083
Val Loss: 1.6629 Acc: 0.3333	Val Loss: 2.0962 Acc: 0.4000
Epoch 5/20	Epoch 15/20
Train Loss: 0.8916 Acc: 0.6167	Train Loss: 0.5911 Acc: 0.8000
Val Loss: 1.7076 Acc: 0.3333	Val Loss: 1.8195 Acc: 0.3333
Epoch 6/20	Epoch 16/20
Train Loss: 0.7504 Acc: 0.6417	Train Loss: 0.5990 Acc: 0.6833
Val Loss: 1.3035 Acc: 0.4000	Val Loss: 1.2196 Acc: 0.6667
Epoch 7/20	Epoch 17/20
Train Loss: 0.7153 Acc: 0.6917	Train Loss: 0.5742 Acc: 0.7500
Val Loss: 1.1550 Acc: 0.5333	Val Loss: 1.0857 Acc: 0.6667
Epoch 8/20	Epoch 18/20
Train Loss: 0.7390 Acc: 0.6417	Train Loss: 0.6219 Acc: 0.7583
Val Loss: 1.2244 Acc: 0.4667	Val Loss: 1.7722 Acc: 0.5333
Epoch 9/20	Epoch 19/20
Train Loss: 0.6846 Acc: 0.7250	Train Loss: 0.5844 Acc: 0.7917
Val Loss: 1.1568 Acc: 0.4000	Val Loss: 1.2895 Acc: 0.4667
Epoch 10/20	Epoch 20/20
Train Loss: 0.6145 Acc: 0.7667	Train Loss: 0.4762 Acc: 0.8250
Val Loss: 1.1130 Acc: 0.4000	Val Loss: 1.2953 Acc: 0.4667
	Test Loss: 1.3104 Acc: 0.6667

- Versi Modifikasi dengan menambah dropout sebesar 0.2:** Pada penambahan dropout 0.2, terlihat bahwa model mengalami peningkatan dalam akurasi pada data pelatihan dari 30.00% menjadi 69.17%, menunjukkan adanya peningkatan performa yang signifikan. Namun, performa pada data validasi tetap fluktuatif dengan akurasi yang berubah-ubah antara 20.00% hingga 60.00%. Hal ini menunjukkan bahwa model masih belum sepenuhnya mampu menggeneralisasi dengan baik pada data baru, terutama mengingat fluktuasi yang cukup besar pada akurasi data validasi. Pada data uji, model mencapai akurasi sebesar 33.33% dengan loss sebesar 4.0925.

```

Epoch 1/10
Train Loss: 2.0564 Acc: 0.3000
Val Loss: 1.1580 Acc: 0.4000
Epoch 2/10
Train Loss: 1.2372 Acc: 0.3667
Val Loss: 17.3230 Acc: 0.2000
Epoch 3/10
Train Loss: 1.0529 Acc: 0.4250
Val Loss: 15.1383 Acc: 0.2667
Epoch 4/10
Train Loss: 1.0248 Acc: 0.4333
Val Loss: 7.6346 Acc: 0.3333
Epoch 5/10
Train Loss: 0.9110 Acc: 0.5333
Val Loss: 1.8492 Acc: 0.5333
Epoch 6/10
Train Loss: 0.8586 Acc: 0.6000
Val Loss: 1.2400 Acc: 0.6000
Epoch 7/10
Train Loss: 0.7761 Acc: 0.6833
Val Loss: 1.9365 Acc: 0.4667
Epoch 8/10
Train Loss: 0.8750 Acc: 0.6083
Val Loss: 1.6180 Acc: 0.4000
Epoch 9/10
Train Loss: 0.7354 Acc: 0.6583
Val Loss: 1.3269 Acc: 0.4000
Epoch 10/10
Train Loss: 0.7322 Acc: 0.6917
Val Loss: 1.6967 Acc: 0.2667
Test Loss: 4.0925 Acc: 0.3333

```

- Versi Modifikasi dengan menambah dropout sebesar 0.5:** Pada penambahan dropout 0.5, terlihat bahwa model mengalami fluktuasi yang lebih besar dalam performa. Meskipun terdapat peningkatan pada akurasi data pelatihan dari 32.50% menjadi 60.00%, namun performa pada data validasi masih fluktuatif dengan akurasi yang berubah-ubah antara 26.67% hingga 60.00%. Hal ini menunjukkan bahwa penambahan dropout 0.5 belum memberikan hasil yang signifikan dalam mengurangi overfitting, bahkan terkadang malah menyebabkan fluktuasi yang lebih besar dalam performa model. Pada data uji, model mencapai akurasi sebesar 40.00% dengan loss sebesar 3.5699.

```

Epoch 1/10
Train Loss: 1.7475 Acc: 0.3250
Val Loss: 1.8330 Acc: 0.4000
Epoch 2/10
Train Loss: 1.4185 Acc: 0.3417
Val Loss: 2.3110 Acc: 0.3333
Epoch 3/10
Train Loss: 1.2388 Acc: 0.3500
Val Loss: 1.9140 Acc: 0.2667
Epoch 4/10
Train Loss: 1.1032 Acc: 0.4583
Val Loss: 2.1170 Acc: 0.3333
Epoch 5/10
Train Loss: 1.0317 Acc: 0.4750
Val Loss: 4.6151 Acc: 0.4000
Epoch 6/10
Train Loss: 1.0525 Acc: 0.4917
Val Loss: 1.8159 Acc: 0.3333
Epoch 7/10
Train Loss: 0.9380 Acc: 0.5250
Val Loss: 2.8798 Acc: 0.3333
Epoch 8/10
Train Loss: 0.8934 Acc: 0.6167
Val Loss: 5.1762 Acc: 0.3333
Epoch 9/10
Train Loss: 0.8282 Acc: 0.6500
Val Loss: 6.5551 Acc: 0.4000
Epoch 10/10
Train Loss: 0.7993 Acc: 0.6000
Val Loss: 6.6669 Acc: 0.6000
Test Loss: 3.5699 Acc: 0.4000

```

- Versi Modifikasi dengan menambah dropout sebesar 0.2 dan menambah jumlah epoch:** Pada penambahan dropout 0.2 dan epoch menjadi 20, terjadi fluktuasi yang cukup besar pada performa model. Meskipun terdapat peningkatan akurasi data pelatihan dari 36.67% menjadi 80.00%, namun performa pada data validasi masih fluktuatif dengan akurasi yang berubah-ubah antara 20.00% hingga 53.33%. Hal ini menunjukkan bahwa penambahan dropout 0.2 belum memberikan hasil yang signifikan dalam mengurangi overfitting, terutama pada data validasi. Pada data uji, model mencapai akurasi sebesar 46.67% dengan loss sebesar 1.8201, yang menunjukkan bahwa performa model masih perlu ditingkatkan.



Epoch 1/20	Epoch 11/20
Train Loss: 1.9586 Acc: 0.3667	Train Loss: 0.7765 Acc: 0.6250
Val Loss: 9.6378 Acc: 0.3333	Val Loss: 1.4078 Acc: 0.4667
Epoch 2/20	Epoch 12/20
Train Loss: 1.1035 Acc: 0.4417	Train Loss: 0.7739 Acc: 0.5917
Val Loss: 150.3564 Acc: 0.2667	Val Loss: 1.9601 Acc: 0.4667
Epoch 3/20	Epoch 13/20
Train Loss: 1.1029 Acc: 0.4750	Train Loss: 0.7083 Acc: 0.7000
Val Loss: 69.0040 Acc: 0.2667	Val Loss: 1.4859 Acc: 0.3333
Epoch 4/20	Epoch 14/20
Train Loss: 1.0480 Acc: 0.5000	Train Loss: 0.7048 Acc: 0.6833
Val Loss: 13.1653 Acc: 0.2000	Val Loss: 1.1024 Acc: 0.3333
Epoch 5/20	Epoch 15/20
Train Loss: 0.9373 Acc: 0.5750	Train Loss: 0.7482 Acc: 0.6917
Val Loss: 1.2751 Acc: 0.3333	Val Loss: 1.2023 Acc: 0.3333
Epoch 6/20	Epoch 16/20
Train Loss: 0.8464 Acc: 0.5833	Train Loss: 0.6769 Acc: 0.6500
Val Loss: 1.4500 Acc: 0.4000	Val Loss: 1.2893 Acc: 0.4000
Epoch 7/20	Epoch 17/20
Train Loss: 0.7648 Acc: 0.6250	Train Loss: 0.6425 Acc: 0.7250
Val Loss: 1.3986 Acc: 0.4000	Val Loss: 1.6193 Acc: 0.3333
Epoch 8/20	Epoch 18/20
Train Loss: 0.7838 Acc: 0.6167	Train Loss: 0.6312 Acc: 0.7417
Val Loss: 1.4808 Acc: 0.5333	Val Loss: 1.2304 Acc: 0.4000
Epoch 9/20	Epoch 19/20
Train Loss: 0.8061 Acc: 0.6417	Train Loss: 0.6765 Acc: 0.7083
Val Loss: 1.1886 Acc: 0.4667	Val Loss: 1.2866 Acc: 0.4667
Epoch 10/20	Epoch 20/20
Train Loss: 0.7039 Acc: 0.7167	Train Loss: 0.5837 Acc: 0.8000
Val Loss: 1.0831 Acc: 0.5333	Val Loss: 1.3623 Acc: 0.4000
	Test Loss: 1.8201 Acc: 0.4667

- Versi Modifikasi dengan menambah dropout sebesar 0.5 dan menambah jumlah epoch:** Peningkatan dropout menjadi 0.5 dan epoch menjadi 20 menghasilkan fluktuasi yang signifikan pada performa model. Meskipun terjadi peningkatan akurasi data pelatihan dari 30.00% menjadi 75.83%, namun performa pada data validasi masih fluktuatif dengan akurasi yang bervariasi antara 20.00% hingga 53.33%. Hal ini menunjukkan bahwa peningkatan dropout belum memberikan dampak yang signifikan dalam mengurangi overfitting. Pada data uji, model mencapai akurasi sebesar 33.33% dengan loss yang cukup tinggi, yaitu 12.4216, menunjukkan bahwa performa model perlu ditingkatkan terutama dalam meningkatkan akurasi pada data validasi dan mengurangi fluktuasi performa pada data pelatihan dan validasi.

Epoch 1/20	Epoch 11/20
Train Loss: 1.5305 Acc: 0.3000	Train Loss: 0.8023 Acc: 0.6083
Val Loss: 1.1689 Acc: 0.3333	Val Loss: 11.1490 Acc: 0.4000
Epoch 2/20	Epoch 12/20
Train Loss: 1.2339 Acc: 0.3667	Train Loss: 0.7568 Acc: 0.6750
Val Loss: 2.4847 Acc: 0.3333	Val Loss: 12.2533 Acc: 0.4667
Epoch 3/20	Epoch 13/20
Train Loss: 1.1391 Acc: 0.4583	Train Loss: 0.7696 Acc: 0.6583
Val Loss: 1.2364 Acc: 0.2000	Val Loss: 11.0345 Acc: 0.4667
Epoch 4/20	Epoch 14/20
Train Loss: 1.0722 Acc: 0.3917	Train Loss: 0.6968 Acc: 0.6917
Val Loss: 2.9144 Acc: 0.3333	Val Loss: 9.6479 Acc: 0.4667
Epoch 5/20	Epoch 15/20
Train Loss: 1.1191 Acc: 0.4250	Train Loss: 0.7428 Acc: 0.7000
Val Loss: 5.5449 Acc: 0.3333	Val Loss: 10.4476 Acc: 0.4667
Epoch 6/20	Epoch 16/20
Train Loss: 1.0603 Acc: 0.4167	Train Loss: 0.8894 Acc: 0.6000
Val Loss: 6.7114 Acc: 0.4000	Val Loss: 3.8050 Acc: 0.4000
Epoch 7/20	Epoch 17/20
Train Loss: 0.9478 Acc: 0.5167	Train Loss: 0.7249 Acc: 0.6417
Val Loss: 7.5436 Acc: 0.4667	Val Loss: 3.2460 Acc: 0.5333
Epoch 8/20	Epoch 18/20
Train Loss: 0.9000 Acc: 0.5667	Train Loss: 0.6815 Acc: 0.6750
Val Loss: 8.5013 Acc: 0.4667	Val Loss: 3.2651 Acc: 0.4667
Epoch 9/20	Epoch 19/20
Train Loss: 0.8309 Acc: 0.6167	Train Loss: 0.6433 Acc: 0.6917
Val Loss: 11.8427 Acc: 0.4667	Val Loss: 7.0669 Acc: 0.4667
Epoch 10/20	Epoch 20/20
Train Loss: 0.7938 Acc: 0.6167	Train Loss: 0.6399 Acc: 0.7583
Val Loss: 11.4210 Acc: 0.4667	Val Loss: 9.9248 Acc: 0.4667
	Test Loss: 12.4216 Acc: 0.3333

### 3. Model 3

- Versi Asli:** Terlihat bahwa model mengalami peningkatan akurasi pada data pelatihan dari 25.83% menjadi 74.17%. Namun, performa pada data validasi masih fluktuatif dengan akurasi berkisar antara 13.33% hingga 46.67%, menunjukkan bahwa model masih mengalami overfitting. Pada data uji, model mencapai akurasi 53.33% dengan loss sebesar 0.9485. Loss pada data pelatihan menunjukkan penurunan yang konsisten, tetapi loss pada data validasi tidak menunjukkan pola penurunan yang stabil, tetap berada di sekitar 1.4136 pada akhir epoch.

```

Epoch 1/10
Loss Pelatihan: 13.1751 Akurasi: 0.2583
Loss Validasi: 7.1258 Akurasi: 0.3333
Epoch 2/10
Loss Pelatihan: 4.0261 Akurasi: 0.3833
Loss Validasi: 1.2627 Akurasi: 0.2667
Epoch 3/10
Loss Pelatihan: 1.0121 Akurasi: 0.4750
Loss Validasi: 1.1221 Akurasi: 0.2000
Epoch 4/10
Loss Pelatihan: 1.0572 Akurasi: 0.4333
Loss Validasi: 1.1270 Akurasi: 0.2000
Epoch 5/10
Loss Pelatihan: 1.0476 Akurasi: 0.4417
Loss Validasi: 1.1105 Akurasi: 0.1333
Epoch 6/10
Loss Pelatihan: 1.0032 Akurasi: 0.5500
Loss Validasi: 1.0768 Akurasi: 0.3333
Epoch 7/10
Loss Pelatihan: 0.9079 Akurasi: 0.6333
Loss Validasi: 1.1683 Akurasi: 0.2000
Epoch 8/10
Loss Pelatihan: 0.8090 Akurasi: 0.6000
Loss Validasi: 1.2151 Akurasi: 0.2667
Epoch 9/10
Loss Pelatihan: 0.6880 Akurasi: 0.6750
Loss Validasi: 1.3362 Akurasi: 0.2000
Epoch 10/10
Loss Pelatihan: 0.5862 Akurasi: 0.7417
Loss Validasi: 1.4136 Akurasi: 0.4667
Loss Uji: 0.9485 Akurasi: 0.5333

```

- **Versi Modifikasi dengan menambah jumlah epoch:** Terlihat bahwa model mengalami peningkatan akurasi pada data pelatihan dari 28.33% menjadi 100.00%. Namun, performa pada data validasi masih fluktuatif dengan akurasi berkisar antara 26.67% hingga 46.67%, menunjukkan bahwa model masih mengalami overfitting. Pada data uji, model mencapai akurasi 60.00% dengan loss sebesar 1.7047. Loss pada data pelatihan menunjukkan penurunan yang konsisten, tetapi loss pada data validasi tidak menunjukkan pola penurunan yang stabil, bahkan meningkat pada akhir epoch.

Epoch 1/20	Epoch 11/20
Loss Pelatihan: 11.4736 Akurasi: 0.2833	Loss Pelatihan: 0.7609 Akurasi: 0.6583
Loss Validasi: 5.8499 Akurasi: 0.3333	Loss Validasi: 1.1254 Akurasi: 0.2667
Epoch 2/20	Epoch 12/20
Loss Pelatihan: 2.7535 Akurasi: 0.4333	Loss Pelatihan: 0.6932 Akurasi: 0.7500
Loss Validasi: 1.2209 Akurasi: 0.3333	Loss Validasi: 1.1796 Akurasi: 0.2667
Epoch 3/20	Epoch 13/20
Loss Pelatihan: 1.0511 Akurasi: 0.4417	Loss Pelatihan: 0.5768 Akurasi: 0.8083
Loss Validasi: 1.1647 Akurasi: 0.3333	Loss Validasi: 1.2146 Akurasi: 0.3333
Epoch 4/20	Epoch 14/20
Loss Pelatihan: 1.0292 Akurasi: 0.4417	Loss Pelatihan: 0.4829 Akurasi: 0.8750
Loss Validasi: 1.0918 Akurasi: 0.4000	Loss Validasi: 1.3372 Akurasi: 0.3333
Epoch 5/20	Epoch 15/20
Loss Pelatihan: 1.0283 Akurasi: 0.5000	Loss Pelatihan: 0.3608 Akurasi: 0.9250
Loss Validasi: 1.0594 Akurasi: 0.3333	Loss Validasi: 1.2454 Akurasi: 0.4667
Epoch 6/20	Epoch 16/20
Loss Pelatihan: 1.0237 Akurasi: 0.4667	Loss Pelatihan: 0.2850 Akurasi: 0.9417
Loss Validasi: 1.0530 Akurasi: 0.4000	Loss Validasi: 1.5049 Akurasi: 0.4667
Epoch 7/20	Epoch 17/20
Loss Pelatihan: 1.0083 Akurasi: 0.4750	Loss Pelatihan: 0.1938 Akurasi: 0.9833
Loss Validasi: 1.0690 Akurasi: 0.3333	Loss Validasi: 1.7922 Akurasi: 0.3333
Epoch 8/20	Epoch 18/20
Loss Pelatihan: 0.9600 Akurasi: 0.5167	Loss Pelatihan: 0.1335 Akurasi: 0.9833
Loss Validasi: 1.0770 Akurasi: 0.2667	Loss Validasi: 1.8227 Akurasi: 0.4000
Epoch 9/20	Epoch 19/20
Loss Pelatihan: 0.9029 Akurasi: 0.6417	Loss Pelatihan: 0.0970 Akurasi: 0.9833
Loss Validasi: 1.0807 Akurasi: 0.2667	Loss Validasi: 1.9898 Akurasi: 0.4000
Epoch 10/20	Epoch 20/20
Loss Pelatihan: 0.8548 Akurasi: 0.6583	Loss Pelatihan: 0.0479 Akurasi: 1.0000
Loss Validasi: 1.1373 Akurasi: 0.3333	Loss Validasi: 2.4110 Akurasi: 0.2667
	Loss Uji: 1.7047 Akurasi: 0.6000

- Versi Modifikasi dengan menambah dropout sebesar 0.2:** Pada penambahan dropout 0.2, terlihat bahwa model mengalami peningkatan akurasi pada data pelatihan dari 34.17% menjadi 84.17%. Namun, performa pada data validasi masih fluktuatif dengan akurasi berkisar antara 26.67% hingga 53.33%, menunjukkan bahwa model masih mengalami overfitting. Pada data uji, model mencapai akurasi 53.33% dengan loss sebesar 1.0220. Loss pada data pelatihan menunjukkan penurunan yang konsisten, tetapi loss pada data validasi tidak menunjukkan pola penurunan yang stabil, tetap berada di sekitar 1.0374 pada akhir epoch.

```
Epoch 1/10
Loss Pelatihan: 13.2373 Akurasi: 0.3417
Loss Validasi: 5.2502 Akurasi: 0.3333
Epoch 2/10
Loss Pelatihan: 3.5033 Akurasi: 0.3917
Loss Validasi: 1.3548 Akurasi: 0.3333
Epoch 3/10
Loss Pelatihan: 1.3225 Akurasi: 0.4167
Loss Validasi: 1.0880 Akurasi: 0.4000
Epoch 4/10
Loss Pelatihan: 0.9709 Akurasi: 0.4917
Loss Validasi: 1.0377 Akurasi: 0.4667
Epoch 5/10
Loss Pelatihan: 0.9589 Akurasi: 0.5333
Loss Validasi: 1.0596 Akurasi: 0.2667
Epoch 6/10
Loss Pelatihan: 0.8402 Akurasi: 0.6417
Loss Validasi: 1.0625 Akurasi: 0.4000
Epoch 7/10
Loss Pelatihan: 0.7571 Akurasi: 0.7333
Loss Validasi: 1.0148 Akurasi: 0.4000
Epoch 8/10
Loss Pelatihan: 0.5997 Akurasi: 0.7917
Loss Validasi: 1.1148 Akurasi: 0.4000
Epoch 9/10
Loss Pelatihan: 0.5022 Akurasi: 0.7917
Loss Validasi: 1.1095 Akurasi: 0.5333
Epoch 10/10
Loss Pelatihan: 0.3779 Akurasi: 0.8417
Loss Validasi: 1.0374 Akurasi: 0.5333
Loss Uji: 1.0220 Akurasi: 0.5333
```

- **Versi Modifikasi dengan menambah dropout sebesar 0.5:** Pada penambahan dropout 0.5, terlihat bahwa model mengalami peningkatan akurasi pada data pelatihan dari 33.33% menjadi 61.67%. Namun, performa pada data validasi masih fluktuatif dengan akurasi berkisar antara 20.00% hingga 33.33%, menunjukkan bahwa model masih mengalami overfitting. Pada data uji, model mencapai akurasi 46.67% dengan loss sebesar 1.2701. Loss pada data pelatihan menunjukkan penurunan yang konsisten, tetapi loss pada data validasi tidak menunjukkan pola penurunan yang stabil, tetap berada di sekitar 1.0631 pada akhir epoch.

```

Epoch 1/10
Loss Pelatihan: 10.6997 Akurasi: 0.3333
Loss Validasi: 4.3884 Akurasi: 0.2667
Epoch 2/10
Loss Pelatihan: 3.1398 Akurasi: 0.4667
Loss Validasi: 1.1665 Akurasi: 0.2667
Epoch 3/10
Loss Pelatihan: 0.9957 Akurasi: 0.4750
Loss Validasi: 1.1148 Akurasi: 0.3333
Epoch 4/10
Loss Pelatihan: 1.0652 Akurasi: 0.4750
Loss Validasi: 1.1005 Akurasi: 0.3333
Epoch 5/10
Loss Pelatihan: 1.0622 Akurasi: 0.4667
Loss Validasi: 1.1088 Akurasi: 0.3333
Epoch 6/10
Loss Pelatihan: 1.0453 Akurasi: 0.4833
Loss Validasi: 1.1714 Akurasi: 0.2000
Epoch 7/10
Loss Pelatihan: 1.0520 Akurasi: 0.5583
Loss Validasi: 1.1311 Akurasi: 0.3333
Epoch 8/10
Loss Pelatihan: 0.9814 Akurasi: 0.6250
Loss Validasi: 1.0830 Akurasi: 0.3333
Epoch 9/10
Loss Pelatihan: 0.9413 Akurasi: 0.5667
Loss Validasi: 1.0575 Akurasi: 0.2667
Epoch 10/10
Loss Pelatihan: 0.8462 Akurasi: 0.6167
Loss Validasi: 1.0631 Akurasi: 0.3333
Loss Uji: 1.2701 Akurasi: 0.4667

```

- Versi Modifikasi dengan menambah dropout sebesar 0.2 dan menambah jumlah epoch:** Pada penambahan dropout 0.2 dan epoch menjadi 20, model mengalami peningkatan akurasi pada data pelatihan dari 31.67% menjadi 99.17%. Namun, performa pada data validasi masih fluktuatif dengan akurasi berkisar antara 26.67% hingga 53.33%, menunjukkan bahwa model masih mengalami overfitting. Pada data uji, model mencapai akurasi 73.33% dengan loss sebesar 0.9134. Loss pada data pelatihan menunjukkan penurunan yang konsisten, tetapi loss pada data validasi tidak menunjukkan pola penurunan yang stabil, tetap berada di sekitar 1.2738 pada akhir epoch.

Epoch 1/20  
Loss Pelatihan: 16.9167 Akurasi: 0.3167  
Loss Validasi: 9.6686 Akurasi: 0.3333  
Epoch 2/20  
Loss Pelatihan: 6.1830 Akurasi: 0.4083  
Loss Validasi: 1.3231 Akurasi: 0.2667  
Epoch 3/20  
Loss Pelatihan: 1.0971 Akurasi: 0.4083  
Loss Validasi: 1.0841 Akurasi: 0.3333  
Epoch 4/20  
Loss Pelatihan: 1.0097 Akurasi: 0.4917  
Loss Validasi: 1.0628 Akurasi: 0.4000  
Epoch 5/20  
Loss Pelatihan: 0.9919 Akurasi: 0.4917  
Loss Validasi: 1.0852 Akurasi: 0.3333  
Epoch 6/20  
Loss Pelatihan: 0.9622 Akurasi: 0.4917  
Loss Validasi: 1.0664 Akurasi: 0.3333  
Epoch 7/20  
Loss Pelatihan: 0.8926 Akurasi: 0.6083  
Loss Validasi: 1.1521 Akurasi: 0.4000  
Epoch 8/20  
Loss Pelatihan: 0.8175 Akurasi: 0.6583  
Loss Validasi: 1.1488 Akurasi: 0.4000  
Epoch 9/20  
Loss Pelatihan: 0.6845 Akurasi: 0.7250  
Loss Validasi: 1.1228 Akurasi: 0.3333  
Epoch 10/20  
Loss Pelatihan: 0.6279 Akurasi: 0.7583  
Loss Validasi: 1.1096 Akurasi: 0.4000

```

Epoch 11/20
Loss Pelatihan: 0.4550 Akurasi: 0.8500
Loss Validasi: 1.2835 Akurasi: 0.5333
Epoch 12/20
Loss Pelatihan: 0.4158 Akurasi: 0.8583
Loss Validasi: 1.5332 Akurasi: 0.4000
Epoch 13/20
Loss Pelatihan: 0.3027 Akurasi: 0.9250
Loss Validasi: 1.1673 Akurasi: 0.4000
Epoch 14/20
Loss Pelatihan: 0.2180 Akurasi: 0.9500
Loss Validasi: 1.1750 Akurasi: 0.4667
Epoch 15/20
Loss Pelatihan: 0.1406 Akurasi: 0.9500
Loss Validasi: 1.2937 Akurasi: 0.3333
Epoch 16/20
Loss Pelatihan: 0.1177 Akurasi: 0.9750
Loss Validasi: 1.8771 Akurasi: 0.4667
Epoch 17/20
Loss Pelatihan: 0.0866 Akurasi: 0.9750
Loss Validasi: 1.7992 Akurasi: 0.4000
Epoch 18/20
Loss Pelatihan: 0.0701 Akurasi: 0.9917
Loss Validasi: 1.5684 Akurasi: 0.4667
Epoch 19/20
Loss Pelatihan: 0.0638 Akurasi: 0.9917
Loss Validasi: 1.5556 Akurasi: 0.4667
Epoch 20/20
Loss Pelatihan: 0.0466 Akurasi: 0.9917
Loss Validasi: 1.2738 Akurasi: 0.5333
Loss Uji: 0.9134 Akurasi: 0.7333

```

- **Versi Modifikasi dengan menambah dropout sebesar 0.5 dan menambah jumlah epoch:** Peningkatan dropout menjadi 0.5 dan epoch menjadi 20 menunjukkan peningkatan performa yang signifikan pada data pelatihan, namun performa pada data validasi tidak stabil dan menunjukkan indikasi overfitting. Pada data uji, model mencapai akurasi 53.33% dengan loss sebesar 1.1332.



Epoch 1/20  
Loss Pelatihan: 13.2285 Akurasi: 0.3250  
Loss Validasi: 5.3778 Akurasi: 0.4000  
Epoch 2/20  
Loss Pelatihan: 5.6873 Akurasi: 0.4417  
Loss Validasi: 1.1076 Akurasi: 0.4667  
Epoch 3/20  
Loss Pelatihan: 1.0669 Akurasi: 0.5000  
Loss Validasi: 1.0726 Akurasi: 0.4000  
Epoch 4/20  
Loss Pelatihan: 1.0695 Akurasi: 0.4500  
Loss Validasi: 1.0917 Akurasi: 0.3333  
Epoch 5/20  
Loss Pelatihan: 1.0770 Akurasi: 0.4417  
Loss Validasi: 1.0867 Akurasi: 0.3333  
Epoch 6/20  
Loss Pelatihan: 1.0667 Akurasi: 0.4250  
Loss Validasi: 1.0690 Akurasi: 0.3333  
Epoch 7/20  
Loss Pelatihan: 0.9989 Akurasi: 0.4917  
Loss Validasi: 1.0231 Akurasi: 0.4000  
Epoch 8/20  
Loss Pelatihan: 0.9944 Akurasi: 0.5583  
Loss Validasi: 1.0313 Akurasi: 0.2667  
Epoch 9/20  
Loss Pelatihan: 0.9379 Akurasi: 0.5333  
Loss Validasi: 1.0280 Akurasi: 0.3333  
Epoch 10/20  
Loss Pelatihan: 0.9126 Akurasi: 0.6167  
Loss Validasi: 1.0786 Akurasi: 0.3333

```
Epoch 11/20
Loss Pelatihan: 0.8230 Akurasi: 0.6500
Loss Validasi: 1.1565 Akurasi: 0.2000
Epoch 12/20
Loss Pelatihan: 0.8094 Akurasi: 0.6083
Loss Validasi: 1.1189 Akurasi: 0.2000
Epoch 13/20
Loss Pelatihan: 0.7300 Akurasi: 0.6833
Loss Validasi: 1.0424 Akurasi: 0.4000
Epoch 14/20
Loss Pelatihan: 0.7219 Akurasi: 0.7167
Loss Validasi: 1.1747 Akurasi: 0.2667
Epoch 15/20
Loss Pelatihan: 0.6993 Akurasi: 0.7083
Loss Validasi: 1.2080 Akurasi: 0.2667
Epoch 16/20
Loss Pelatihan: 0.6309 Akurasi: 0.7500
Loss Validasi: 1.0500 Akurasi: 0.4000
Epoch 17/20
Loss Pelatihan: 0.5817 Akurasi: 0.7333
Loss Validasi: 1.1538 Akurasi: 0.2667
Epoch 18/20
Loss Pelatihan: 0.5111 Akurasi: 0.7833
Loss Validasi: 1.3483 Akurasi: 0.2667
Epoch 19/20
Loss Pelatihan: 0.4969 Akurasi: 0.7583
Loss Validasi: 1.0826 Akurasi: 0.4000
Epoch 20/20
Loss Pelatihan: 0.3812 Akurasi: 0.8667
Loss Validasi: 1.0656 Akurasi: 0.4000
Loss Uji: 1.1332 Akurasi: 0.5333
```

#### 4. Kesimpulan

Dalam mengevaluasi hasil eksperimen dengan teknik transfer learning, **model FixEfficientNet-B2 versi asli** menunjukkan kinerja dan performa terbaik dibandingkan model lainnya, dengan dengan akurasi validasi yang sangat tinggi dan akurasi pengujian yang kuat pada 86.67%. Kinerja ini menunjukkan bahwa model tersebut memiliki kapasitas yang baik untuk belajar dari data pelatihan dan menggeneralisasi ke data validasi dan pengujian, meskipun adanya sedikit tanda overfitting, yang terlihat dari perbedaan antara akurasi pelatihan dan pengujian. Hal ini menandakan bahwa model mampu menerapkan apa yang telah dipelajari pada situasi baru, tetapi masih perlu penyesuaian lebih lanjut untuk mengurangi overfitting.

Teknik scratch atau desain dari awal yang paling berhasil adalah **Model 1 yang telah dimodifikasi dengan menambahkan dropout dan menambah jumlah epoch**. Sebelum penyetelan hyperparameter, model tersebut mungkin mengalami masalah overfitting atau underfitting yang menghambat performanya. Namun, dengan penambahan dropout, model menjadi lebih robust dan mampu menggeneralisasi lebih baik terhadap data yang belum

pernah dilihat sebelumnya. Penambahan jumlah epoch juga memungkinkan model untuk melakukan lebih banyak iterasi pembelajaran, yang dapat meningkatkan akurasi model.

Hasil akurasi sebesar 80.00% pada data uji menunjukkan bahwa penyetelan hyperparameter tersebut efektif dalam meningkatkan performa model. Hal ini menggambarkan bahwa dengan teknik desain yang tepat dan penyetelan hyperparameter yang cermat, model dari awal dapat mencapai tingkat akurasi yang tinggi.

Meskipun demikian, pada akhirnya, teknik transfer learning lebih unggul. Transfer learning memanfaatkan pengetahuan yang telah dipelajari oleh model pada dataset yang lebih besar dan kompleks sebelumnya. Dengan menggunakan model yang telah terlatih pada dataset yang besar, model transfer learning dapat belajar fitur-fitur yang lebih umum dan mewakili, yang kemudian meningkatkan kemampuannya untuk menggeneralisasi pada data baru. Selain itu, transfer learning juga menghemat waktu dan sumber daya, karena tidak perlu melatih model dari awal pada dataset yang besar. Konsistensi performa yang lebih tinggi juga merupakan salah satu keunggulan transfer learning, karena model yang menggunakan transfer learning cenderung memiliki performa yang lebih konsisten pada dataset baru.

## 5. **Referensi**

Papers with Code. (n.d.). *Image Classification on ImageNet - State of the Art*. Retrieved from <https://paperswithcode.com/sota/image-classification-on-imagenet?metric=Number%20of%20params>.

TensorFlow. (n.d.). *Image Classification*. Retrieved from <https://www.tensorflow.org/tutorials/images/classification>.

Bitgrit Data Science Publication. (2020, February 11). *Building an Image Classification Model with PyTorch from Scratch*. Retrieved from <https://medium.com/bitgrit-data-science-publication/building-an-image-classification-model-with-pytorch-from-scratch-f10452073212>.

Karuneshu. (2020, July 2). *How to ResNet in PyTorch*. Retrieved from <https://medium.com/@karuneshu21/how-to-resnet-in-pytorch-9acb01f36cf5>.

Keras Team. (n.d.). *MobileNet*. Keras. Retrieved from <https://keras.io/api/applications/mobilenet/>.