



---

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

Tugas: **Tugas Eksplorasi**  
Tanggal: **29 September 2025**

1. **Joshua Palti Sinaga (122140141)**
  2. **Rustian Afencius Marbun (122140155)**
- 

## Eksperimen Arsitektur ResNet-34

### 1 Tahap 1: Analisis Performa Model Dasar (Plain Network)

#### 1.1 Arsitektur Plain-34

Plain-34 adalah arsitektur yang mirip dengan ResNet-34 tanpa skip connection, memiliki 34 layer untuk memproses gambar 224x224 pixels. Arsitektur dimulai dengan convolutional layer 7x7 (stride 2, 64 channel), batch normalization, ReLU, dan max pooling 3x3 (stride 2). Kemudian terdapat empat stage: Stage 1 (3 block, 64 channel), Stage 2 (4 block, 128 channel), Stage 3 (6 block, 256 channel), dan Stage 4 (3 block, 512 channel), dengan stride 2 pada block pertama di stage 2-4. Setiap Plain Block terdiri dari dua layer Conv 3x3 + BatchNorm + ReLU. Arsitektur diakhiri dengan global average pooling dan fully connected layer untuk 5 kelas. Perbedaan utama dengan ResNet-34 adalah output block langsung diproses tanpa dijumlahkan dengan input (tidak ada skip connection).

#### 1.2 Konfigurasi Pelatihan

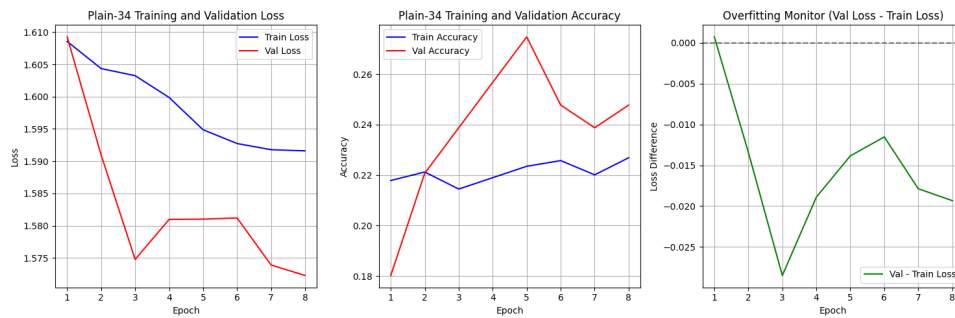
Konfigurasi pelatihan untuk Plain-34 menggunakan learning rate 0.001 (1e-3) dengan optimizer SGD. Batch size untuk training adalah 32, sedangkan untuk validation adalah 64. Model dilatih selama 8 epoch menggunakan CrossEntropyLoss sebagai loss function.

#### 1.3 Implementasi

Implementasi menggunakan PyTorch dengan PlainBlock yang terdiri dari dua Conv3x3-BN-ReLU secara berurutan tanpa skip connection. Dataset 5 kelas makanan Indonesia dibagi 80% training dan 20% validation. Model dilatih 8 epoch dengan monitoring loss dan accuracy.

#### 1.4 Hasil Pelatihan

Hasil menunjukkan performa rendah dengan 21.29M parameter. Training loss turun konsisten (1.609→1.591), validation loss stabil di 1.57. Akurasi sangat rendah: training 22.7%, validation 24.7%, best validation 27.1% (epoch 5), hanya sedikit di atas random guessing (20%). Overfitting monitor menunjukkan nilai negatif setelah epoch 2, mengindikasikan model menghafal training data tanpa generalisasi baik.



Gambar 1: Grafik Training dan Validation Plain-34

## 1.5 Analisis

Performa Plain-34 sangat rendah (max 27%) menunjukkan masalah degradasi pada jaringan dalam tanpa skip connection. Dengan 34 layer, gradient mengalami vanishing gradient problem karena perkalian berulang dengan nilai kecil saat backpropagation, menyebabkan layer awal tidak ter-update efektif. Overfitting terlihat dari validation loss lebih rendah dari training loss setelah epoch 2, dan validation accuracy turun setelah puncak di epoch 5.

## 2 Tahap 2: Implementasi Residual Connection (ResNet-34)

### 2.1 Modifikasi Arsitektur

ResNet-34 identik dengan Plain-34 namun menambahkan skip connection pada setiap block. Perbedaan kunci: Plain-34 menggunakan  $\text{output} = \text{ReLU}(\text{Conv2}(\text{ReLU}(\text{Conv1}(x))))$ , sedangkan ResNet-34 menggunakan  $\text{output} = \text{ReLU}(\text{Conv2}(\text{ReLU}(\text{Conv1}(x))) + x)$ . Penambahan input  $x$  (skip connection) menciptakan "jalan pintas" untuk aliran gradient. Residual block menyimpan input sebagai identity, memproses melalui Conv-BN-ReLU-Conv-BN menghasilkan  $F(x)$ , lalu menjumlahkan  $F(x) + x$  sebelum ReLU final. Model hanya perlu belajar residual  $F(x) = H(x) - x$ , bukan transformasi penuh  $H(x)$ . Untuk dimensi berbeda, digunakan downsample layer (Conv  $1 \times 1$ ). Semua konfigurasi lain sama dengan Plain-34 untuk perbandingan yang adil.

### 2.2 Implementasi

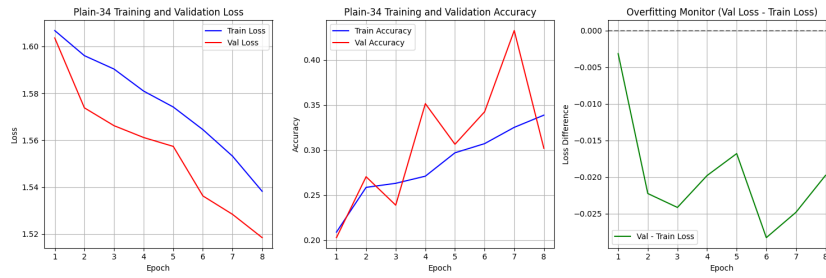
Implementasi ResNetBlock memodifikasi forward function dengan menyimpan input sebagai identity, memproses melalui Conv-BN-ReLU-Conv-BN, lalu menjumlahkan hasil dengan identity (skip connection:  $\text{out} + \text{identity}$ ) sebelum ReLU final. Downsample digunakan jika dimensi berbeda.

### 2.3 Hasil Pelatihan

ResNet-34 menunjukkan peningkatan signifikan dengan parameter sama (21.29M). Loss turun smooth: training  $1.605 \rightarrow 1.538$ , validation  $1.605 \rightarrow 1.516$ . Akurasi meningkat drastis: training 34.4%, validation 30.0%, best validation 44.7% (epoch 6), jauh melampaui Plain-34. Learning curve konsisten naik hingga epoch 6. Overfitting lebih terkontrol dengan loss difference stabil, menunjukkan skip connection membantu regularisasi implisit.

### 2.4 Perbandingan dengan Plain-34

Skip connection memberikan dampak signifikan. Best validation accuracy meningkat dari 27.1%  $\rightarrow$  44.7%, final validation accuracy +5.3%, training accuracy +11.7%. Loss turun: validation -0.057, training -0.053. Yang penting, peningkatan ini dicapai dengan parameter sama persis (21.29M), membuktikan



Gambar 2: Grafik Training dan Validation ResNet-34

efektivitas murni dari skip connection. ResNet-34 menunjukkan konvergensi lebih cepat, kurva smooth dan stabil, berbeda dengan Plain-34 yang stagnan dan fluktuatif.

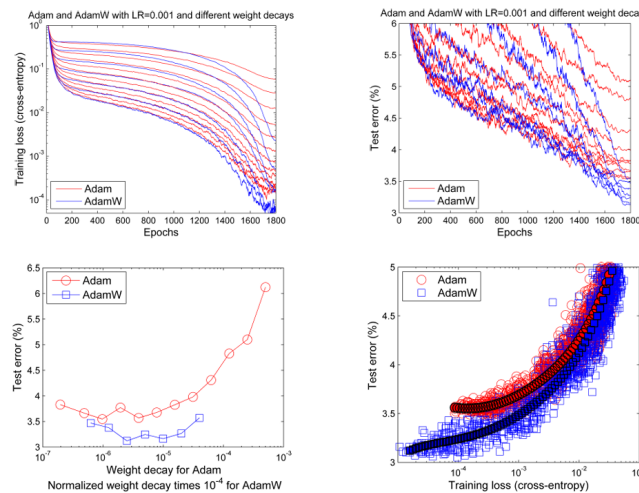
### 3 Tahap 3: Eksperimen Modifikasi Arsitektur ResNet-34

#### 3.1 Pemilihan Modifikasi

Kami memilih dua modifikasi pada arsitektur ResNet-34, yaitu penggantian optimizer Adam ke AdamW dengan weight decay  $5e-4$ , dan arsitektur ResNeXt.

#### 3.2 Justifikasi Pemilihan

AdamW dipilih berdasarkan penelitian Loshchilov dan Hutter [1] yang membuktikan generalisasi lebih baik dari Adam melalui pemisahan weight decay dari gradient update. Weight decay  $5e-4$  memberikan regularisasi efektif tanpa menghambat pembelajaran.



Gambar 3: Perbandingan Adam vs AdamW (Sumber: Loshchilov &amp; Hutter [1])

ResNeXt dipilih berdasarkan paper Xie et al. [2] yang memperkenalkan cardinality sebagai dimensi baru. Grouped convolution membagi channel menjadi kelompok independen, meningkatkan kapasitas model tanpa menambah parameter signifikan. Cardinality 16 dipilih berdasarkan rekomendasi paper untuk keseimbangan optimal.

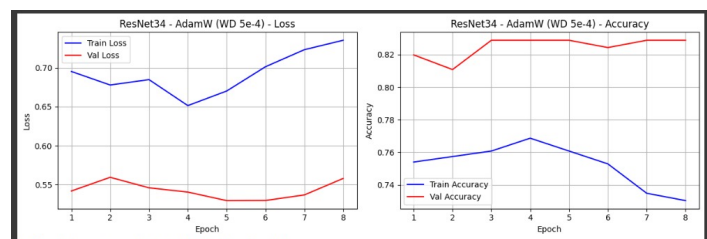
### 3.3 Modifikasi 1: Optimizer AdamW

#### 3.3.1 Implementasi

Optimizer diubah dari Adam ke AdamW (weight decay  $5e-4$ ) dengan learning rate  $3e-4$ , batch size 32/64, dan 8 epoch. AdamW memisahkan weight decay dari gradient update, berbeda dengan L2 regularization pada Adam standar.

#### 3.3.2 Hasil Pelatihan

Best validation accuracy 82.88%, meningkat 38.2 poin dari baseline (44.7%). Metrik F1 0.8245, precision 0.8410, recall 0.8288 menunjukkan performa seimbang. Konvergensi smooth dengan overfitting terkontrol membuktikan efektivitas weight decay sebagai regularisasi.



Gambar 4: Grafik Training dan Validation ResNet-34 dengan AdamW

#### 3.3.3 Analisis

Peningkatan dari 44.7%→82.88% membuktikan dampak pemisahan weight decay dari gradient update. AdamW mencegah weight decay tereduksi oleh adaptive learning rate, menghasilkan regularisasi lebih efektif dan optimisasi lebih stabil.

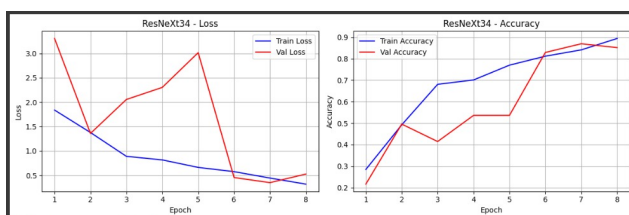
### 3.4 Modifikasi 2: Arsitektur ResNeXt

#### 3.4.1 Implementasi

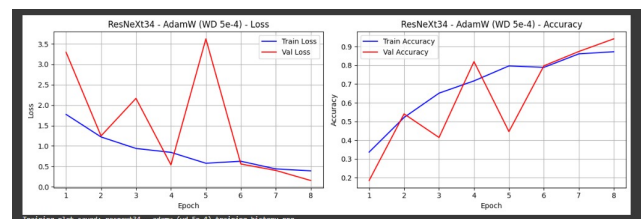
ResNet block diganti dengan ResNeXt-like block (grouped convolution, cardinality 16). Input dibagi 16 kelompok independen, diproses Conv 3x3, lalu diagregat sebelum skip connection. Diuji dengan Adam (LR  $1e-3$ ) dan AdamW (LR  $1e-3$ , WD  $5e-4$ ), batch size 32/64, 8 epoch.

#### 3.4.2 Hasil Pelatihan

ResNeXt+Adam: 86.94% accuracy (+42.2 poin), F1 0.8684, precision 0.8858, recall 0.8694. ResNeXt+AdamW (terbaik): 94.14% accuracy, F1 0.9428, precision 0.9509, recall 0.9414.



(a) ResNeXt34 dengan Adam



(b) ResNeXt34 dengan AdamW

Gambar 5: Perbandingan Grafik Training ResNeXt34

### 3.4.3 Analisis

Grouped convolution (16 jalur paralel) meningkatkan kapasitas representasi melalui ekstraksi fitur lebih beragam. Peningkatan dari 44.7%→86.94% membuktikan efektivitas arsitektur. Kombinasi ResNeXt+AdamW (94.14%) menunjukkan sinergi optimal, konsisten dengan temuan Xie et al. [2].

## 3.5 Perbandingan Modifikasi

### 3.5.1 ResNet-34: Adam vs AdamW

Penggantian Adam ke AdamW meningkatkan akurasi 38.2 poin (hampir 2x lipat). Peningkatan recall > precision menunjukkan identifikasi kelas lebih baik, memvalidasi temuan paper [1] tentang pentingnya pemisahan weight decay.

Tabel 1: Perbandingan ResNet-34 dengan Adam vs AdamW

Metrik	Adam	AdamW	Peningkatan
Best Val Accuracy	44.7%	82.88%	+38.2%
F1 Score	0.7129	0.8245	+0.1116
Precision	0.7947	0.8410	+0.0463
Recall	0.7297	0.8288	+0.0991

### 3.5.2 ResNeXt-34: Adam vs AdamW

AdamW meningkatkan akurasi 7.2 poin (86.94%→94.14%), membuktikan efektivitas untuk berbagai arsitektur. Peningkatan seimbang precision dan recall menunjukkan generalisasi lebih baik.

Tabel 2: Perbandingan ResNeXt-34 dengan Adam vs AdamW

Metrik	Adam	AdamW	Peningkatan
Best Val Accuracy	86.94%	94.14%	+7.2%
F1 Score	0.8684	0.9428	+0.0744
Precision	0.8858	0.9509	+0.0651
Recall	0.8694	0.9414	+0.0720

### 3.5.3 Dampak Arsitektur ResNet vs ResNeXt (Optimizer Sama)

Dengan optimizer sama (Adam), ResNeXt mengungguli ResNet 42.2 poin (hampir 2x lipat). Peningkatan recall lebih besar membuktikan grouped convolution menangkap variasi kelas lebih efektif, konsisten dengan paper [2].

## 4 Peran dan Kontribusi AI Assistant

Dalam pengerjaan tugas ini, AI Assistant (Google Gemini dan ChatGPT) digunakan untuk membantu implementasi kode dan *brainstorming* terkait resnet34 dan modifikasi yang akan kami lakukan. Untuk detail lengkap prompt dan interaksi dengan AI, dapat dilihat pada link berikut: (1) Google Gemini: <https://g.co/gemini/share/3c7328db60e2>, dan (2) ChatGPT: <https://chatgpt.com/share/68dfb201-d978-8001-b60b-7f91d1ee34a5>.

## References

- [1] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” *arXiv preprint arXiv:1711.05101*, 2017.
- [2] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.