IF3270 Pembelajaran Mesin
Tugas Besar 1
Feedforward Neural Network

Disusun oleh:

Kelompok 8

Ariel Herfrison	13522002
Kristo Anugrah	13522024
Francesco Michael Kusuma	13522038

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

2025/2026

BAB 1
DESKRIPSI PERSOALAN

Perkembangan teknologi *Machine Learning* dan *Deep Learning* dalam beberapa tahun terakhir telah membawa revolusi yang signifikan dalam bidang *pattern recognition* dan *computer vision*. Salah satu algoritma yang dasar adalah *Artificial Neural Network(ANN)* yang terinspirasi oleh cara kerja sistem saraf biologis manusia. ANN bisa mempelajari pola yang kompleks dari data melalui proses pelatihan(training) sehingga bisa digunakan untuk persoalan seperti klasifikasi, prediksi, dan sebagainya.

ANN memiliki beberapa arsitektur, salah satunya adalah *Feedforward Neural Network*(FFNN). Ini adalah model paling dasar dan banyak digunakan. FFNN terdiri dari *input layer*, satu atau lebih *hidden layer*, dan *output layer*. Data yang ada merambat secara searah(*feedforward*) dari input menuju output tanpa adanya siklik. Kemampuan FFNN dalam memodelkan hubungan di berbagai data, baik linear maupun non-linear, antara *input* dan *output* membuatnya sangat cocok untuk berbagai aplikasi.

Salah satu dataset yang terkenal untuk pengujian dalam klasifikasi adalah *MNIST(Modified National Institute of Standards and Technology)*. MNIST terdiri dari 70.000 gambar digit tulisan tangan dalam bentuk *grayscale*. Dataset ini menjadi *benchmark* untuk evaluasi model. Kesederhanaan dan ukurannya yang tidak terlalu besar memungkinkan eksperimen cepat tanpa memerlukan sumber daya komputasi tinggi, sementara kompleksitasnya yang cukup menantang membuatnya relevan untuk menguji kemampuan model.

FFNN memiliki beberapa kelebihan, yaitu:

1. Kesederhanaan arsitektur
   FFNN sangat mudah untuk dipahami dan diimplementasikan sehingga sangat cocok sebagai langkah awal dalam pembelajaran.
2. Kemampuan pembelajaran untuk non-linear separable data
   FFNN memiliki berbagai fungsi aktivasi, seperti ReLU atau sigmoid. Oleh karena itu, FFNN dapat dengan mudah mendapat hubungan kompleks dalam data.

Hyperparameter yang ada pada FFNN meliputi:

1. Jumlah layer
2. Jumlah neuron
3. Fungsi aktivasi
4. Learning rate
5. Regularisasi

BAB 2
PEMBAHASAN

2.1. Implementasi

2.1.1. Deskripsi Kelas

1. ANNScratch

   Kelas ANNScratch ada di file ann.py dan berisi implementasi arsitektur Feedforward Neural Network. Kelas ini memiliki fungsi untuk melakukan *training, prediction, save,* dan *load*. Detail method dan atribut yang ada pada kelas ini adalah sebagai berikut.

Detail Method	
Method	Deskripsi
`def __init__(...)`	Fungsi ini merupakan konstruktor dari kelas ANNScratch. Pada fungsi ini seluruh atribut kelas didefinisikan. Pada fungsi ini juga bobot dan gradien FFNN diinisialisasi.
`def initialize_weights(...)`	Merupakan fungsi pembantu (*helper function*) untuk menginisialisasikan bobot pada layer input dan hidden FFNN. Fungsi ini akan menginisialisasikan bobot berdasarkan opsi yang dimasukkan ke dalam konstruktor.
`def initialize_output_weights(...)`	Merupakan fungsi pembantu (*helper function*) untuk menginisialisasikan bobot pada layer output FFNN. Fungsi ini akan menginisialisasikan bobot berdasarkan opsi yang dimasukkan ke dalam konstruktor. Fungsi ini dipanggil saat training dimulai.
`def safe_exp(...)`	Merupakan fungsi untuk menghitung nilai konstanta e dipangkatkan suatu bilangan real. Fungsi ini menerima suatu *array-like* dan menghitung nilai $e\text{^}x$ dari setiap x pada *array* tersebut. Fungsi ini dibuat karena tendensi kerentanan fungsi eksponen untuk overflow.
`def activation(...)`	Merupakan fungsi yang menghitung nilai fungsi aktivasi dari sebuah layer. Menerima sebuah *array-like* dan opsi fungsi aktivasi. Mengembalikan *array-like* sebagai hasil dari fungsi aktivasi terhadap input yang diberikan. Fungsi ini digunakan pada saat *forward*

	propagation.
`def activation_ derivative(...)`	Merupakan fungsi yang menghitung nilai turunan dari fungsi aktivasi dari sebuah layer. Menerima sebuah *array-like* dan opsi fungsi aktivasi. Mengembalikan *array-like* sebagai hasil dari turunan fungsi aktivasi terhadap input yang diberikan. Fungsi ini digunakan pada saat *backpropagation*.
`def loss_ function(...)`	Merupakan fungsi yang menghitung nilai fungsi loss terhadap input yang diberikan. Menerima sebuah *array-like* dan opsi fungsi loss untuk diterapkan.
`def loss_ gradient(...)`	Merupakan fungsi yang menghitung turunan pertama dari fungsi loss terhadap input yang diberikan. Menerima sebuah *array-like* dan opsi fungsi loss untuk diterapkan.
`def regularization_ loss(...)`	Merupakan fungsi untuk menghitung loss dari regularisasi yang dilakukan. Menerima input opsi regularisasi yaitu L1 atau L2.
`def compute_loss (...)`	Merupakan fungsi untuk menghitung loss, yaitu regularization loss ditambah loss function. Menerima sebuah *array-like* dan mengembalikan nilai loss.
`def apply_rms_norm (...)`	Merupakan fungsi untuk menerapkan normalisasi rms. Menerima sebuah *array-like* dan menerapkan normalisasi rms pada input tersebut.
`def backward(...)`	Melakukan tahap *backpropagation* pada arsitektur FFNN. Fungsi ini sekaligus memanggil tahap *forward propagation*.
`def update_weights (...)`	Merupakan sebuah fungsi pembantu untuk memperbarui bobot serta gradien pada FFNN.
`def fit(...)`	Merupakan fungsi yang dipanggil saat proses training. Fungsi ini akan melakukan training berdasarkan parameter yang diberikan saat inisialisasi.
`def predict(...)`	Melakukan tahap *forward propagation* pada

	FFNN. Mengembalikan hasil akhir, yaitu nilai yang ada pada neuron layer output. Menerima sebuah *array-like* yaitu input pada layer input.
`def get_model_state(...)`	Merupakan fungsi pembantu yang mengembalikan atribut kelas seperti batch_size, banyak layer, fungsi aktivasi, dll.
`def save_model(...)`	Merupakan fungsi yang akan menyimpan model dan juga berbagai parameternya ke dalam sebuah file pickle dengan ekstensi .pkl. File tersebut dapat digunakan untuk me-load model kembali. Fungsi ini juga akan menyimpan bobot serta gradien FFNN saat fungsi dipanggil.
`def load_model(...)`	Merupakan fungsi yang akan me-*load* model dari file .pkl yang telah disimpan. Menerima input berupa path dari file .pkl yang telah disimpan.
`def visualize(...)`	Merupakan fungsi pembantu untuk memvisualisasikan FFNN. Fungsi ini menggunakan kelas ANNVisualizer untuk memvisualisaskan arsitektur dari parameter inisialisasi.
Detail Atribut	
Atribut	Deskripsi
`self.neurons`	Variabel yang menyatakan banyak layer yang ada pada FFNN.
`self.activations`	Merupakan list yang berisi fungsi aktivasi pada setiap layer.
`self.epochs`	Variabel yang menyatakan banyak epoch pada proses training.
`self.loss`	Variabel yang berisi fungsi yang dipakai saat *forward propagation dan backpropagation*.
`self.learning_rate`	Variabel learning rate yang akan digunakan saat proses *backpropagation*.
`self.initialization`	Variabel yang berisi metode inisialisasi yang akan digunakan untuk menginisiasi bobot serta gradien pada FFNN.

`self.batch_size`	Variabel yang berisi batch size yang akan digunakan saat proses training.
`self.verbose`	Variabel yang menyatakan status verbose FFNN. Jika bernilai True, proses training akan meng-output loss pada setiap epoch.
`self.regularization`	Variabel yang berisi regularisasi (jika ada) yang akan diterapkan pada FFNN.
`self.reg_lambda`	Variabel regularization lambda. Digunakan ketika ada regularisasi yang diterapkan ketika proses regularisasi.
`self.epsilon`	Variabel yang berisi nilai epsilon. Digunakan untuk menormalisasikan penghitungan fungsi log. Berisi nilai yang sangat kecil (10^-7).
`self.alpha`	Variabel yang digunakan ketika fungsi aktivasi leaky ReLU.
`self.rms_norm`	Variabel yang digunakan jika normalisasi rms ingin dilakukan pada FFNN.
`self.weights`	List yang berisi bobot dari setiap layer. Bobot layer ke-i akan direpresentasikan sebagai numpy array 2 dimensi.
`self.biases`	List yang berisi bias dari setiap layer. Bobot layer ke-i akan direpresentasikan sebagai numpy array 2 dimensi.
`self.normalize_output`	Variabel yang digunakan jika FFNN ingin dinormalisasi outputnya.
`self.initialization_config`	Variabel yang menyimpan config dari inisialisasi instansi ANNScratch.

2. ANNVisualizer
   Kelas ANNVisualizer ada di file visualizer.py dan berisi implementasi dari visualisasi FFNN yang ada pada kelas ANNScratch.

Detail Method	
Method	Deskripsi

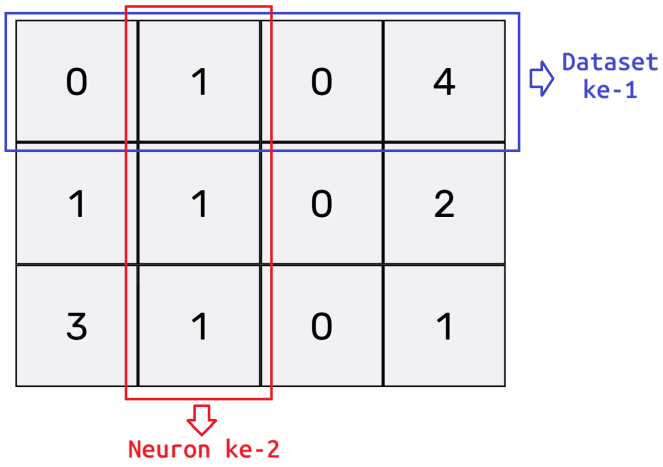
`def __init__(...)`	Fungsi ini merupakan konstruktor dari kelas ANNVisualizer. Pada fungsi ini seluruh atribut kelas didefinisikan. Pada fungsi ini juga graf yang akan ditampilkan diinisialisasi.
`def _build_graph (...)`	Merupakan fungsi yang digunakan untuk membangun graf yang akan ditunjukkan. Fungsi ini akan menggunakan API yang ada pada library networkx.
`def visualize_ network(...)`	Fungsi ini akan memvisualisasikan bobot serta gradien yang ada pada FFNN. Fungsi ini memvisualisasikan kedua graf secara bersamaan *side-by-side*.
`def plot_weight_ distribution (...)`	Fungsi ini memvisualisasikan bobot yang ada dalam suatu layer. Fungsi ini dapat memvisualisasikan seluruh layer atau layer tertentu.
`def plot_neuron_ analysis(...)`	Fungsi ini memvisualisasikan informasi yang ada pada suatu neuron, termasuk bobot serta gradien
`def visualize_all (...)`	Fungsi ini adalah fungsi pembantu yang akan memvisualisasikan arsitektur FFNN dan juga bobot dari setiap layer.
Detail Atribut	
Atribut	Deskripsi
`self.model`	Variabel yang menyimpan model ANNScratch yang akan divisualisasikan.
`self.graph`	Merupakan variabel objek DiGraph yang ada pada library networkx.

2.1.2. Penjelasan Forward Propagation

Pada ANNScratch, bobot serta gradien FFNN dalam satu layer disimpan dalam bentuk matriks. Hal ini untuk menyederhanakan kalkulasi pada tahap *forward propagation* menjadi perkalian matriks. Seluruh operasi perkalian matriks pada ANNScratch menggunakan fungsi pada numpy, yang juga akan mempercepat proses kalkulasi.

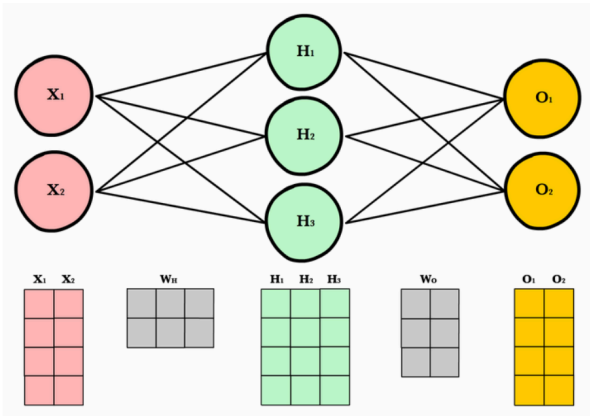
Proses *forward propagation* pada kelas ANNScratch terdapat pada fungsi predict. Fungsi ini menerima sebuah input X yang dianggap sebagai bobot neuron pada layer

input. Misal batch size proses training adalah bs dan banyak neuron pada layer ke-i didenotasikan sebagai $n[i]$. Maka input X harus memiliki dimensi $bs \times n[0]$.



Representasi matriks untuk layer dengan 4 neuron dan batch size 3

Untuk melakukan tahapan *feed forward* dari layer i menuju ke layer i+1, perlu dilakukan perkalian matriks antara output dari layer i dengan bobot layer ke-i. Perlu diingat bahwa bobot pada layer ke-i disimpan dalam bentuk matriks dengan dimensi $n[i] \times n[i+1]$. Ini artinya bobot dari neuron ke-j menuju neuron ke-k di layer berikutnya disimpan pada matriks di indeks (j, k). Perkalian matriks ini akan menghasilkan matriks dengan dimensi $bs \times n[i+1]$ yang merupakan nilai neuron pada layer ke-i untuk setiap data pada batch. Perkalian matriks dilakukan dengan fungsi np.matmul.



Representasi matriks bobot dan juga output pada FFNN. Matriks abu-abu merepresentasikan matriks bobot. Matriks berwarna merepresentasikan output pada layer dengan batch size 4

Proses perkalian matriks tersebut ekuivalen dengan proses penghitungan nilai neuron seperti biasanya. Hal ini dapat dilihat hasil output pada batch ke-j hanya bergantung pada nilai di batch ke-j dan bobot seluruh neuron pada layer sebelumnya. Setelah melakukan perkalian matriks, nilai bias akan ditambahkan pada setiap data pada batch. Perlu diketahui bahwa bias yang menuju layer $i+1$ disimpan sebagai

matriks dengan dimensi $1 \times n[i+1]$. Untuk menambahkan nilai bias pada setiap batch, matriks tersebut perlu di-"ulang" sebanyak batch size untuk menghasilkan matriks $bs \times n[i+1]$. Proses "pengulangan" dilakukan dengan np.tile.
Proses ini akan dilakukan dari layer input hingga didapat matriks output pada layer output. Matriks tersebut dijadikan sebagai hasil prediksi untuk batch tersebut. Perlu diketahui bahwa matriks input dan output pada setiap layer akan disimpan pada sebuah list. Hal ini karena input dan output pada setiap layer akan digunakan pada tahap *backpropagation*.

2.1.3. Penjelasan Backward Propagation dan Weight Update

Tahap backward propagation dimulai dari layer output menuju layer input. Pada layer output, akan diinisialisasi variabel delta yaitu gradien fungsi loss pada nilai prediksi terhadap nilai yang sebenarnya. Perhitungan ini akan menghasilkan matriks dengan dimensi $bs \times n[output]$.

Kemudian, untuk setiap layer, akan dihitung nilai dari turunan fungsi aktivasi terhadap hasil output pada layer tersebut. Perhitungan ini akan menghasilkan matriks dengan dimensi $bs \times n[i]$. Hal ini artinya matriks delta dan matriks turunan aktivasi memiliki ukuran yang sama. Untuk mendapatkan nilai delta pada layer ini, dilakukan *element-wise multiplication* antara matriks delta dengan matriks turunan aktivasi.

Matriks delta kemudian digunakan untuk menghitung nilai gradien pada layer ke i-1. Hal ini dicapai dengan mengalikan input layer ke-i yang di transpose dengan matriks delta. Karena matriks input memiliki dimensi $bs \times n[i-1]$ dan matriks delta memiliki dimensi $bs \times n[i]$, hasil perkalian kedua matriks akan menghasilkan matriks baru dengan dimensi $n[i-1] \times n[i]$, yang cocok dengan bentuk matriks bobot dari layer i-1 menuju i. Perkalian matriks tersebut ekuivalen dengan menghitung nilai δ secara independen untuk setiap data pada batch, dan kemudian menjumlahkannya.

Penghitungan gradien bias serupa dengan penghitungan gradien bobot. Hanya saja, karena nilai bias selalu 1, tidak perlu dilakukan proses perkalian matriks. Mengalikan matriks dengan matriks yang berisi dengan 1 sama saja dengan melakukan penjumlahan secara kolom.

Setelah menghitung gradien bobot dan bias, matriks delta akan diperbarui. Hal ini dilakukan dengan meng-assign delta menjadi delta dikali matriks bobot yang telah di transpos. Matriks delta yang awalnya memiliki dimensi $bs \times n[i]$ dikalikan dengan matriks bobot dengan dimensi $n[i-1] \times n[i]$. Ini artinya perkalian ini akan

menghasilkan matriks dengan dimensi $bs \times n[i-1]$, yang akan menjadi variabel delta di iterasi berikutnya.

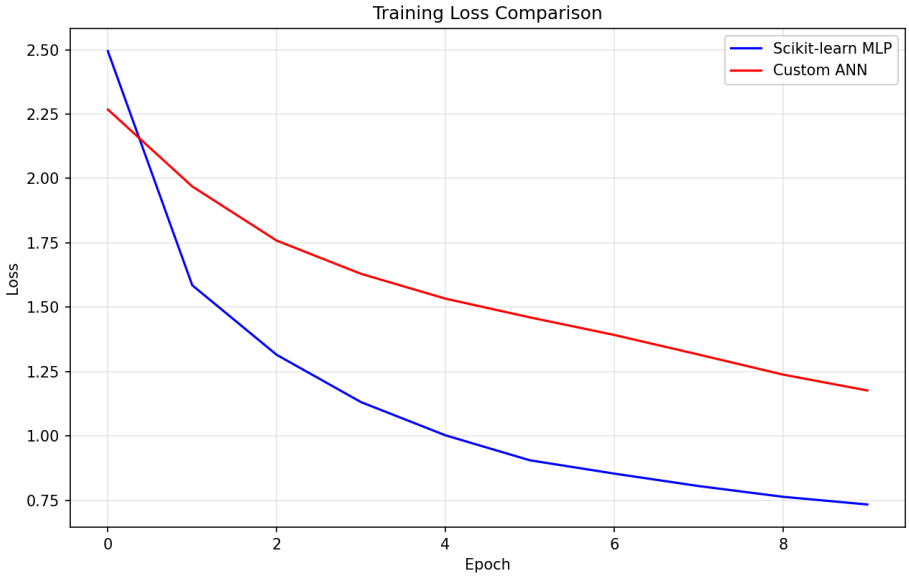
Setelah semua iterasi *backward propagation* selesai, akan dilakukan regularisasi sesuai opsi yang dipilih (jika pengguna memilih menerapkan regularisasi). Setelah itu, akan dilakukan *gradient clipping*. Gradient clipping adalah proses dimana gradien bobot dan bias dibatasi dari -1 hingga 1 untuk mencegah terjadinya *exploding gradient problem*.

2.2. Hasil Pengujian

1. Pengaruh **depth (banyak layer)** dan **width (banyak neuron per layer)**
   a. Variasi depth

Hyperparameters	
Width semua layer	5
Fungsi aktivasi	Sigmoid
Loss	Categorical Cross Entropy
Batch Size	32
Learning Rate	0.01
Epoch	10
Initialization	Uniform: - Lower Bound: -1 - Upper Bound: 1 - Seed: 42

i. Jumlah layer = 3	
	Hasil prediksi - Test Accuracy: 0.7229 - `Test Accuracy: 0.7229`
	Grafik loss



Iterasi Custom ANN:

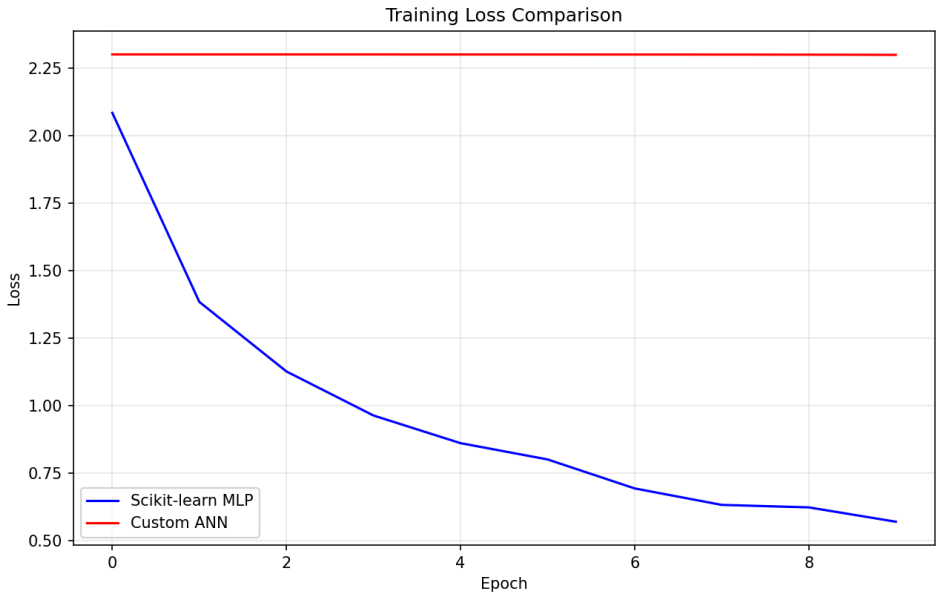
- Epoch 1/10, Loss: 2.2679
- Epoch 2/10, Loss: 1.9689
- Epoch 3/10, Loss: 1.7586
- Epoch 4/10, Loss: 1.6297
- Epoch 5/10, Loss: 1.5332
- Epoch 6/10, Loss: 1.4605
- Epoch 7/10, Loss: 1.3922
- Epoch 8/10, Loss: 1.3161
- Epoch 9/10, Loss: 1.2382
- Epoch 10/10, Loss: 1.1765

ii. Jumlah layer = 5

Hasil prediksi

- Test Accuracy: 0.1126
- `Test Accuracy: 0.1126`

Grafik loss



Iterasi Custom ANN:

- Epoch 1/10, Loss: 2.3014
- Epoch 2/10, Loss: 2.3014
- Epoch 3/10, Loss: 2.3014
- Epoch 4/10, Loss: 2.3013
- Epoch 5/10, Loss: 2.3013
- Epoch 6/10, Loss: 2.3012
- Epoch 7/10, Loss: 2.3011
- Epoch 8/10, Loss: 2.3009
- Epoch 9/10, Loss: 2.3005
- Epoch 10/10, Loss: 2.2998

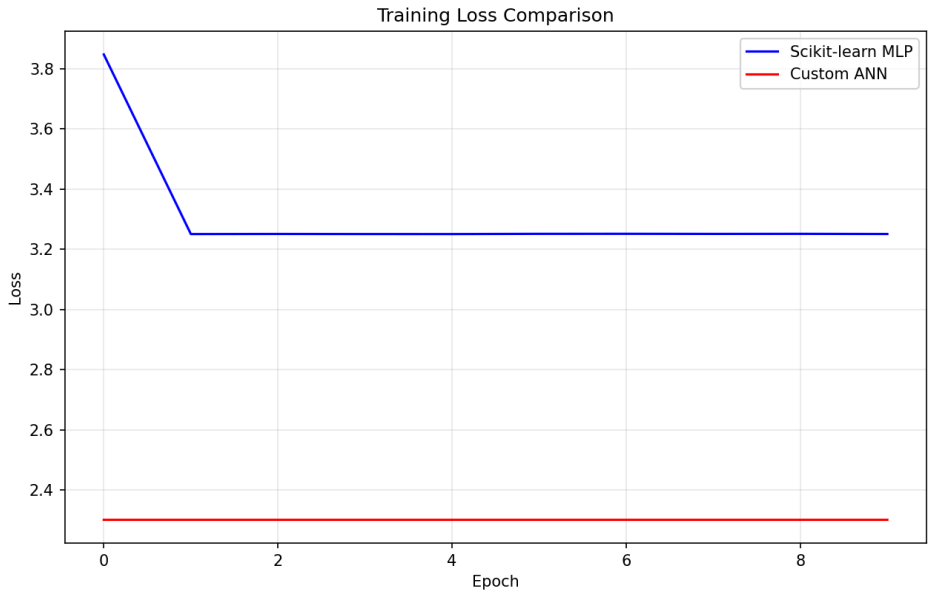
iii. Jumlah layer = 10

Hasil prediksi

- Test Accuracy: 0.1126
-

```
Test Accuracy: 0.1126
```

Grafik loss



Iterasi Custom ANN:

Perbandingan

Dapat dilihat bahwa semakin dalam sebuah FFNN, hasil akurasi dari dataset tes semakin berkurang. Dapat dilihat juga dari grafik bahwa semakin dalam sebuah FFNN, semakin tinggi nilai lossnya. Dari fakta tersebut dapat disimpulkan bahwa semakin dalam sebuah FFNN performa yang dihasilkan semakin buruk.

b. Variasi width

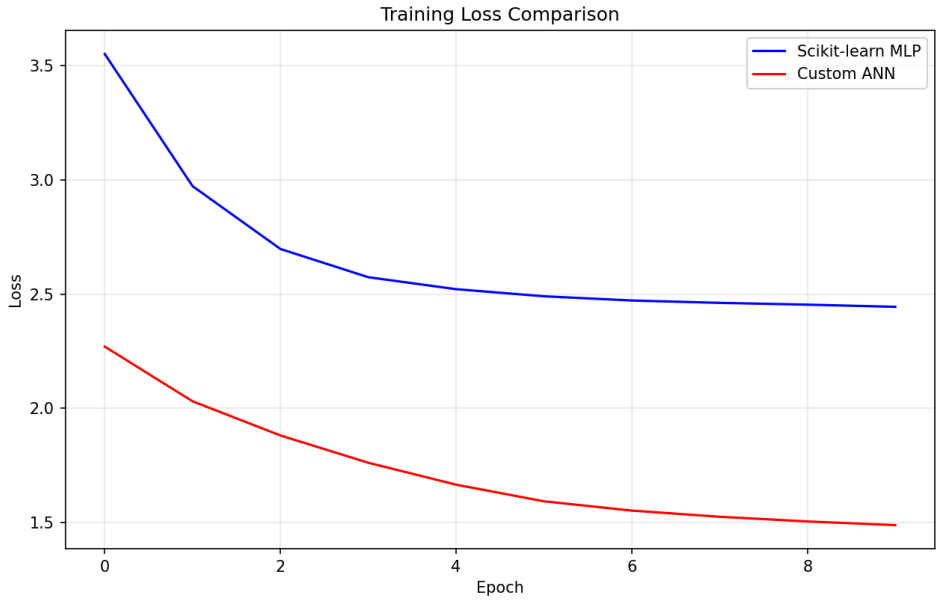
Hyperparameters	
Depth	3
Fungsi aktivasi	Sigmoid
Loss	Categorical Cross Entropy
Batch Size	32
Learning Rate	0.01
Epoch	10
Initialization	Uniform: - Lower Bound: -1 - Upper Bound: 1 - Seed: 42

i. Width semua layer = 3

Hasil prediksi

- Test Accuracy: 0.3520
-
```
Test Accuracy: 0.3520
```

Grafik loss



Iterasi Custom ANN:

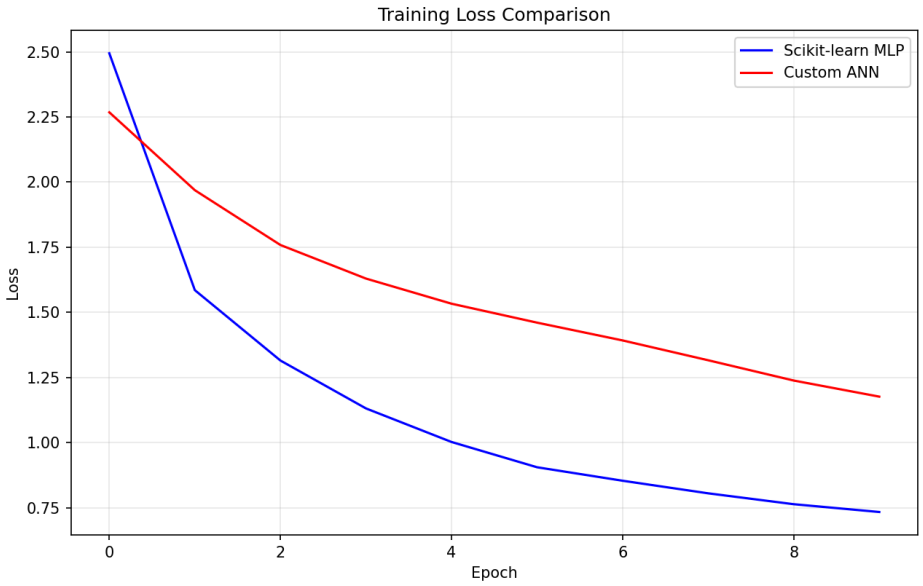
- Epoch 1/10, Loss: 2.2699
- Epoch 2/10, Loss: 2.0306
- Epoch 3/10, Loss: 1.8810
- Epoch 4/10, Loss: 1.7616
- Epoch 5/10, Loss: 1.6656
- Epoch 6/10, Loss: 1.5928
- Epoch 7/10, Loss: 1.5523
- Epoch 8/10, Loss: 1.5252
- Epoch 9/10, Loss: 1.5048
- Epoch 10/10, Loss: 1.4886

ii. Width semua layer = 5

Hasil prediksi

- Test Accuracy: 0.7229
-
```
Test Accuracy: 0.7229
```

Grafik loss



Iterasi Custom ANN:

- Epoch 1/10, Loss: 2.2679
- Epoch 2/10, Loss: 1.9689
- Epoch 3/10, Loss: 1.7586
- Epoch 4/10, Loss: 1.6297
- Epoch 5/10, Loss: 1.5332
- Epoch 6/10, Loss: 1.4605
- Epoch 7/10, Loss: 1.3922
- Epoch 8/10, Loss: 1.3161
- Epoch 9/10, Loss: 1.2382
- Epoch 10/10, Loss: 1.1765

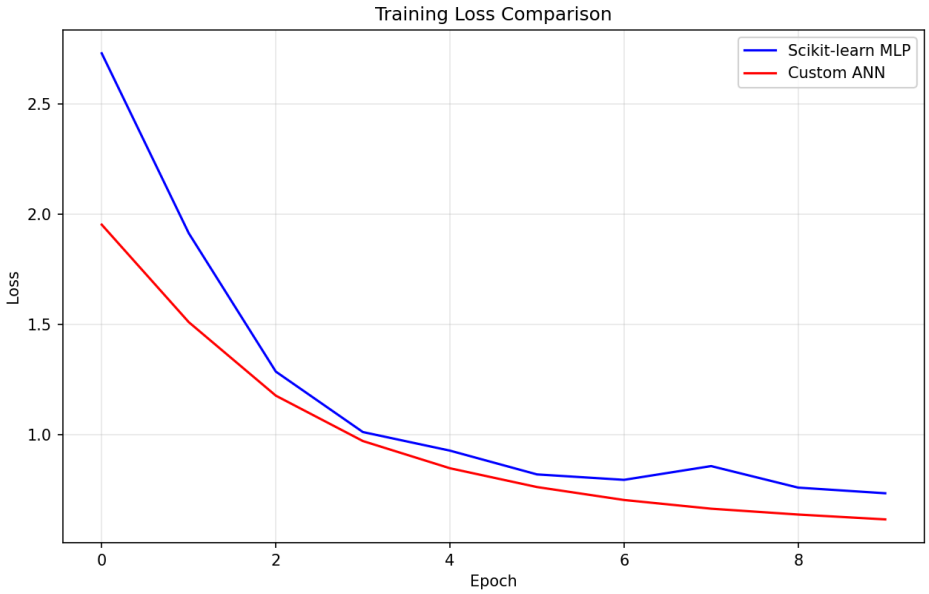
iii. Width semua layer = 10

Hasil prediksi

- Test Accuracy: 0.8332
-

```
Test Accuracy: 0.8332
```

Grafik loss



Iterasi Custom ANN:

- Epoch 1/10, Loss: 1.9534
- Epoch 2/10, Loss: 1.5111
- Epoch 3/10, Loss: 1.1777
- Epoch 4/10, Loss: 0.9718
- Epoch 5/10, Loss: 0.8483
- Epoch 6/10, Loss: 0.7631
- Epoch 7/10, Loss: 0.7044
- Epoch 8/10, Loss: 0.6649
- Epoch 9/10, Loss: 0.6384
- Epoch 10/10, Loss: 0.6164

Perbandingan

Dapat dilihat bahwa semakin lebar sebuah FFNN, hasil akurasi dari dataset tes semakin bertambah. Dapat dilihat juga dari grafik bahwa semakin lebar sebuah FFNN, semakin rendah nilai lossnya. Dari fakta tersebut dapat disimpulkan bahwa semakin lebar sebuah FFNN performa yang dihasilkan semakin baik.

2. Pengaruh **fungsi aktivasi** hidden layer

Hyperparameters	
Depth	3
Width semua layer	10
Fungsi Aktivasi Output	Sigmoid

Loss	Categorical Cross Entropy
Batch Size	32
Learning Rate	0.01
Epoch	10
Initialization	Uniform: - Lower Bound: -1 - Upper Bound: 1 - Seed: 42

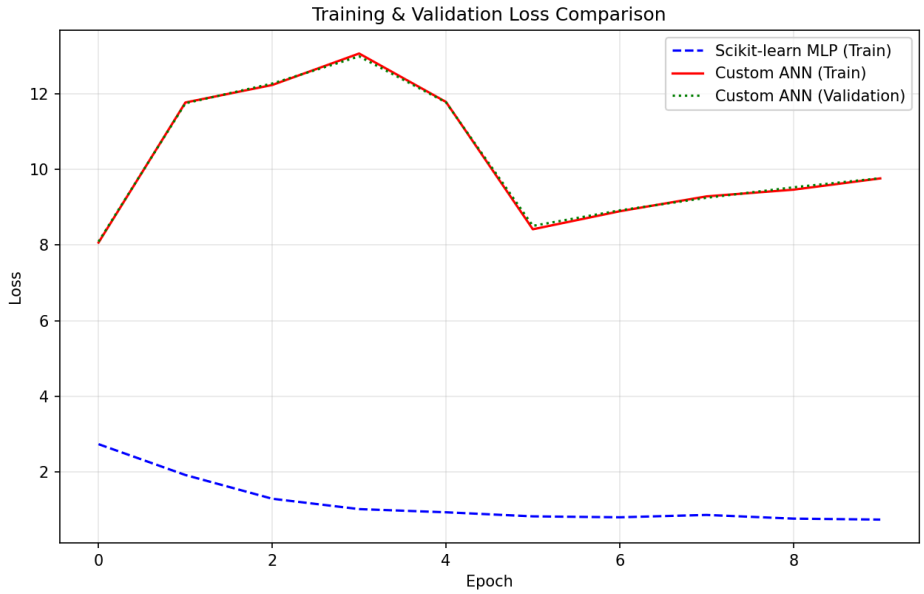
a. Linear

Linear

Hasil prediksi

- Test Accuracy: 0.1379
-
```
Test Accuracy: 0.1379
```

Grafik loss

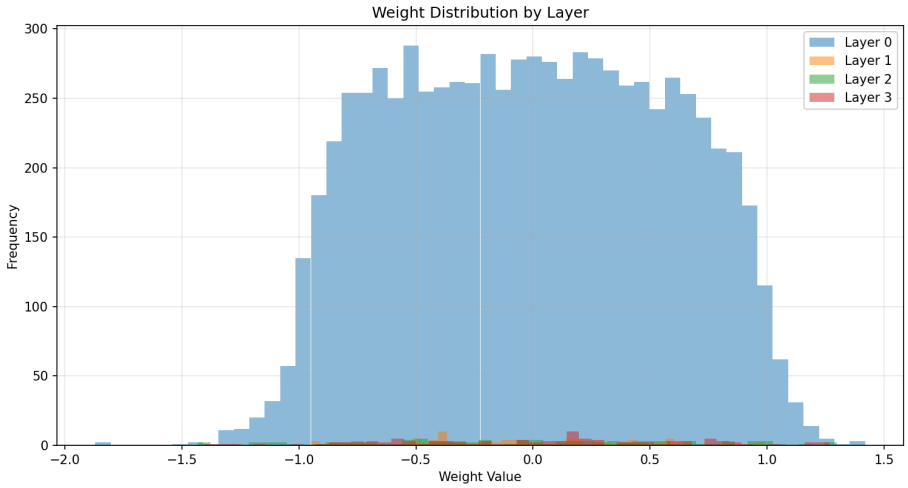


Iterasi Custom ANN:

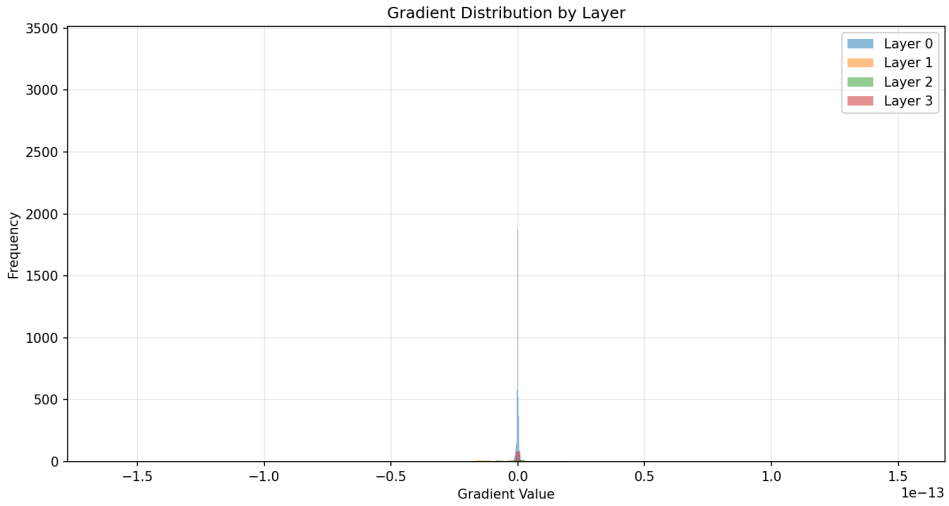
- Epoch 1/10, Training Loss: 8.0667, Validation Loss: 8.0916
- Epoch 2/10, Training Loss: 11.7729, Validation Loss: 11.7517
- Epoch 3/10, Training Loss: 12.2359, Validation Loss: 12.2740
- Epoch 4/10, Training Loss: 13.0690, Validation Loss: 12.9934
- Epoch 5/10, Training Loss: 11.7880, Validation Loss: 11.7774

- Epoch 6/10, Training Loss: 8.4176, Validation Loss: 8.5082
- Epoch 7/10, Training Loss: 8.8946, Validation Loss: 8.9176
- Epoch 8/10, Training Loss: 9.2885, Validation Loss: 9.2515
- Epoch 9/10, Training Loss: 9.4655, Validation Loss: 9.5277
- Epoch 10/10, Training Loss: 9.7635, Validation Loss: 9.7693

Distribusi Bobot:



Distribusi Gradien Bobot:



b. ReLU

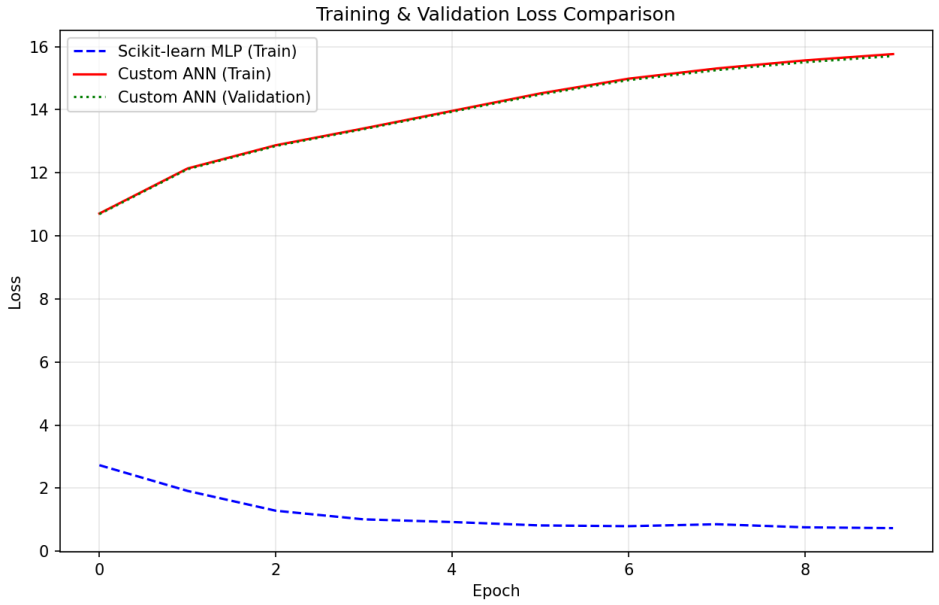
ReLU

Hasil prediksi

- Test Accuracy: 0.1071
-

```
Test Accuracy: 0.1071
```

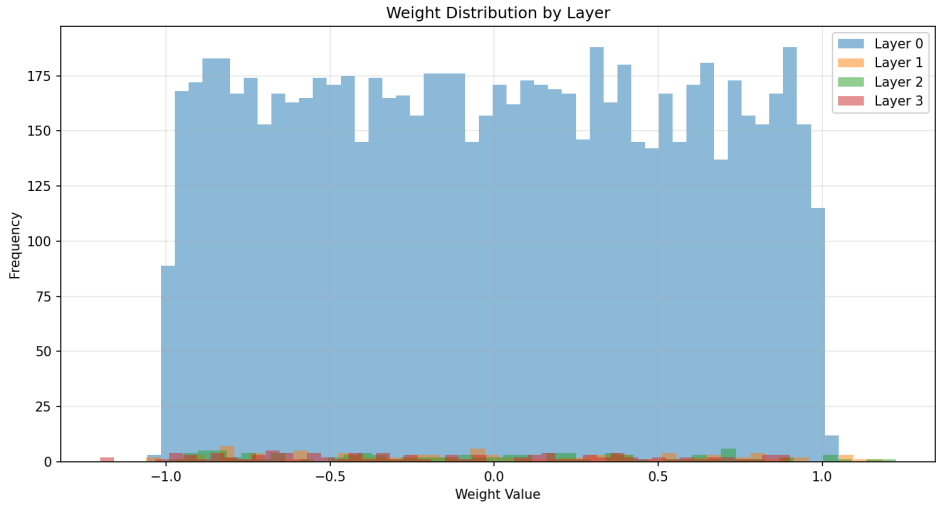
Grafik loss



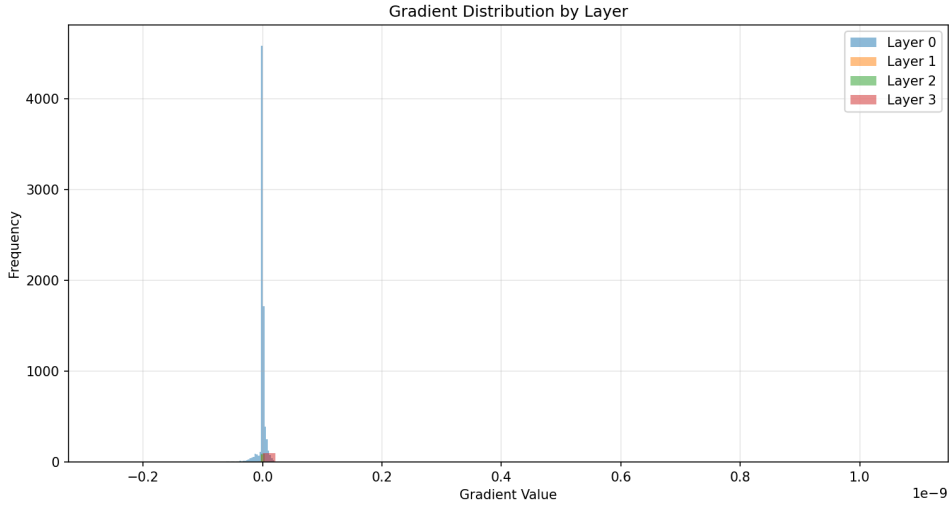
Iterasi Custom ANN:

- Epoch 1/10, Training Loss: 10.7095, Validation Loss: 10.6917
- Epoch 2/10, Training Loss: 12.1380, Validation Loss: 12.1157
- Epoch 3/10, Training Loss: 12.8742, Validation Loss: 12.8511
- Epoch 4/10, Training Loss: 13.4090, Validation Loss: 13.3860
- Epoch 5/10, Training Loss: 13.9632, Validation Loss: 13.9376
- Epoch 6/10, Training Loss: 14.5183, Validation Loss: 14.4837
- Epoch 7/10, Training Loss: 14.9846, Validation Loss: 14.9398
- Epoch 8/10, Training Loss: 15.3086, Validation Loss: 15.2557
- Epoch 9/10, Training Loss: 15.5642, Validation Loss: 15.5072
- Epoch 10/10, Training Loss: 15.7646, Validation Loss: 15.7032

Distribusi Bobot:



Distribusi Gradien Bobot:



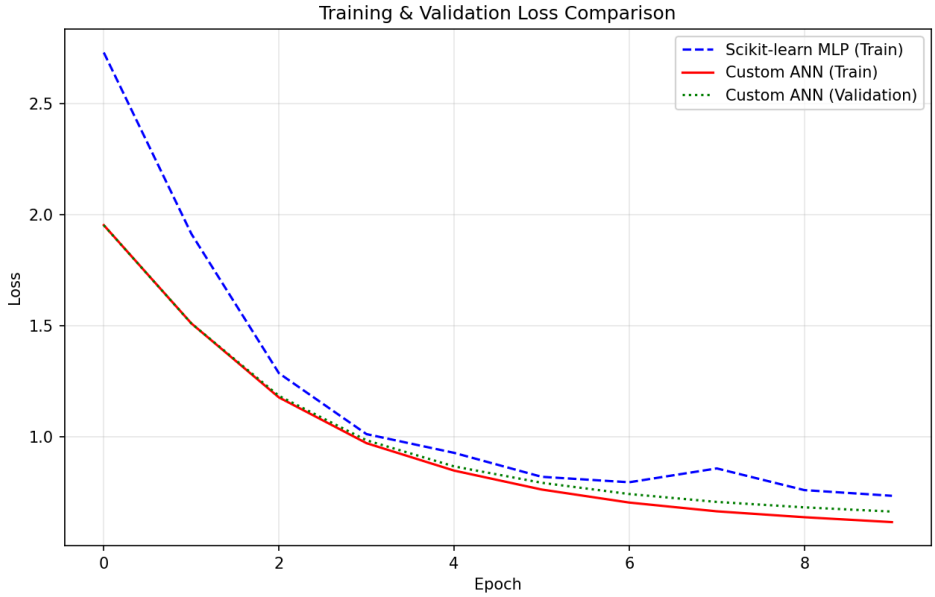
c. Sigmoid

Sigmoid

Hasil prediksi

- Test Accuracy: 0.8332
-
```
Test Accuracy: 0.8332
```

Grafik loss

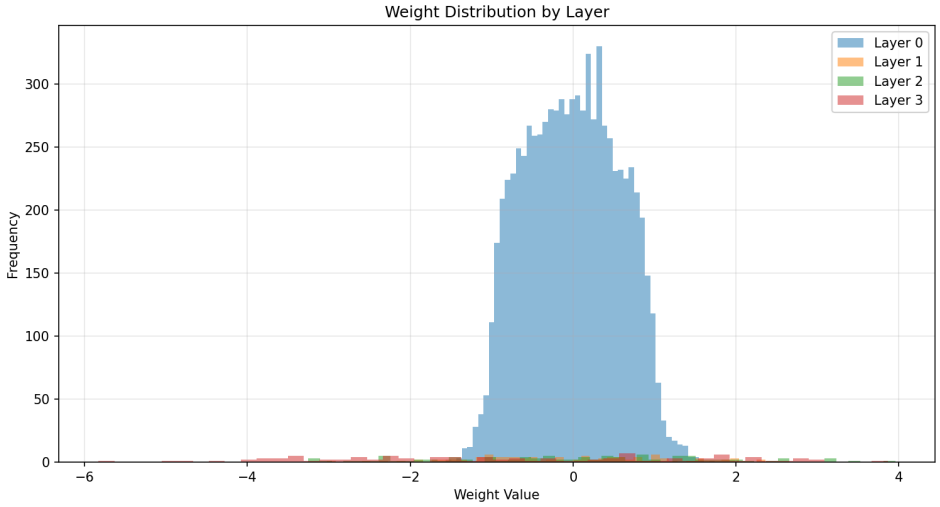


Iterasi Custom ANN:

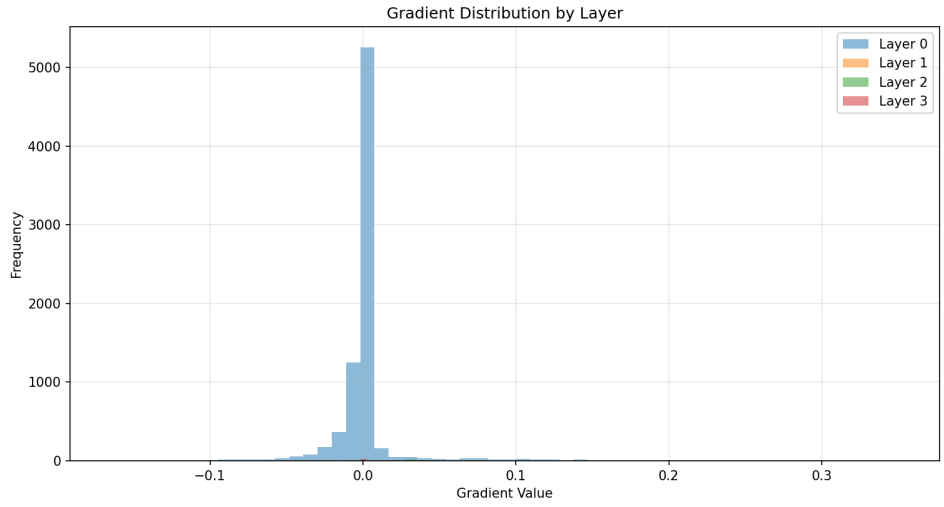
- Epoch 1/10, Training Loss: 1.9534, Validation Loss: 1.9530
- Epoch 2/10, Training Loss: 1.5111, Validation Loss: 1.5108

- Epoch 3/10, Training Loss: 1.1777, Validation Loss: 1.1848
- Epoch 4/10, Training Loss: 0.9718, Validation Loss: 0.9847
- Epoch 5/10, Training Loss: 0.8483, Validation Loss: 0.8672
- Epoch 6/10, Training Loss: 0.7631, Validation Loss: 0.7935
- Epoch 7/10, Training Loss: 0.7044, Validation Loss: 0.7428
- Epoch 8/10, Training Loss: 0.6649, Validation Loss: 0.7072
- Epoch 9/10, Training Loss: 0.6384, Validation Loss: 0.6828
- Epoch 10/10, Training Loss: 0.6164, Validation Loss: 0.6638

Distribusi Bobot:



Distribusi Gradien Bobot:

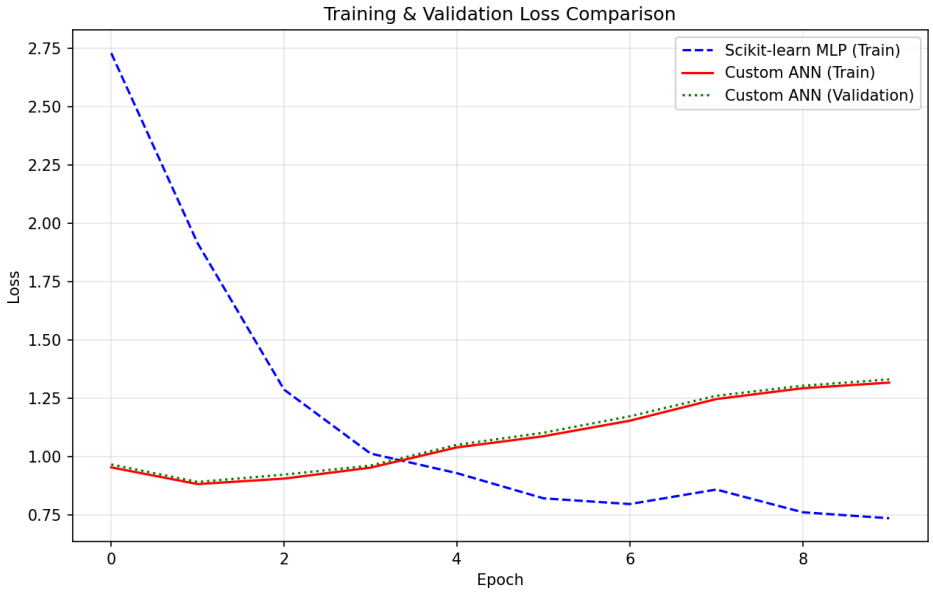


d. Tanh

Tanh	
	Hasil prediksi

- Test Accuracy: 0.6476
-
```
Test Accuracy: 0.6476
```

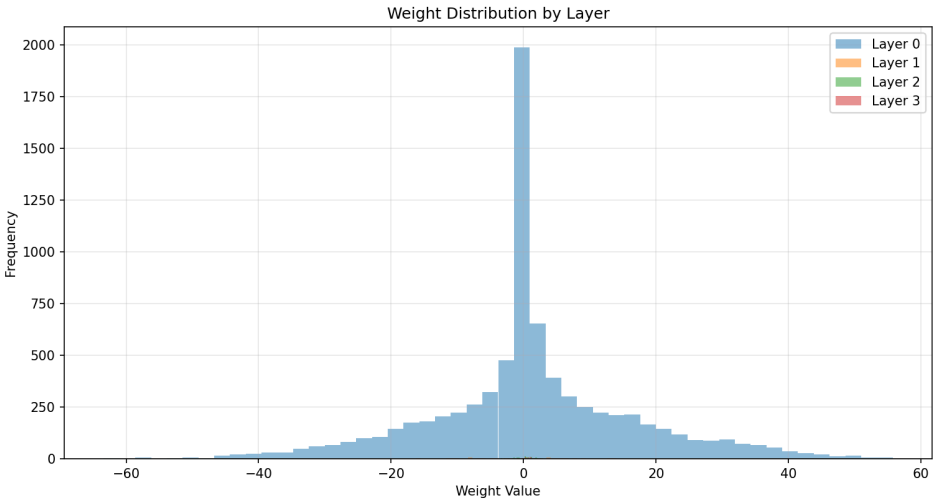
Grafik loss



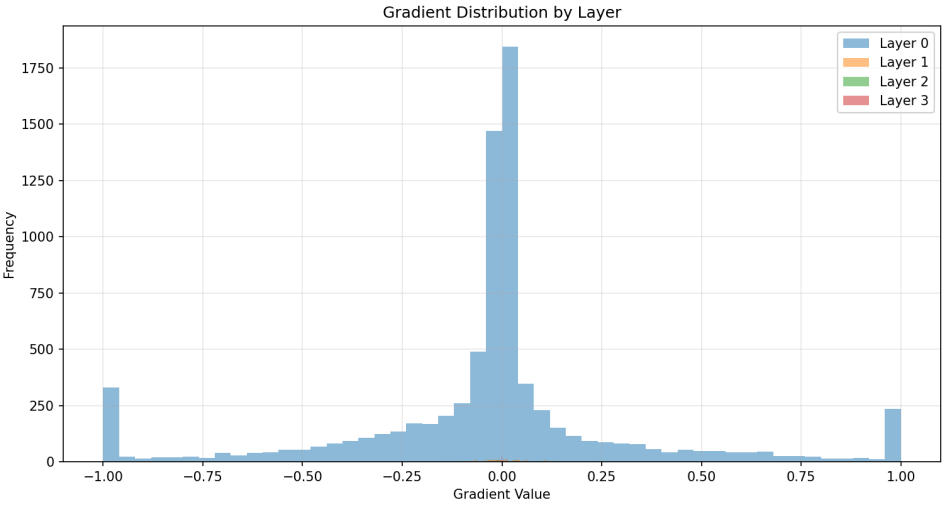
Iterasi Custom ANN:

- Epoch 1/10, Training Loss: 0.9532, Validation Loss: 0.9656
- Epoch 2/10, Training Loss: 0.8815, Validation Loss: 0.8906
- Epoch 3/10, Training Loss: 0.9050, Validation Loss: 0.9221
- Epoch 4/10, Training Loss: 0.9520, Validation Loss: 0.9609
- Epoch 5/10, Training Loss: 1.0389, Validation Loss: 1.0500
- Epoch 6/10, Training Loss: 1.0868, Validation Loss: 1.1018
- Epoch 7/10, Training Loss: 1.1535, Validation Loss: 1.1723
- Epoch 8/10, Training Loss: 1.2463, Validation Loss: 1.2604
- Epoch 9/10, Training Loss: 1.2927, Validation Loss: 1.3035
- Epoch 10/10, Training Loss: 1.3169, Validation Loss: 1.3306

Distribusi Bobot:

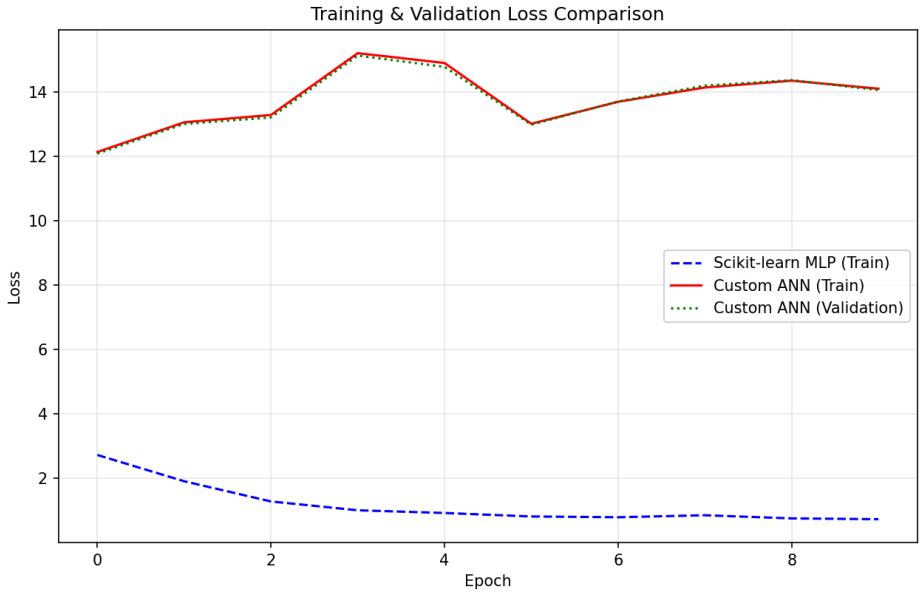


Distribusi Gradien Bobot:



a. Softplus

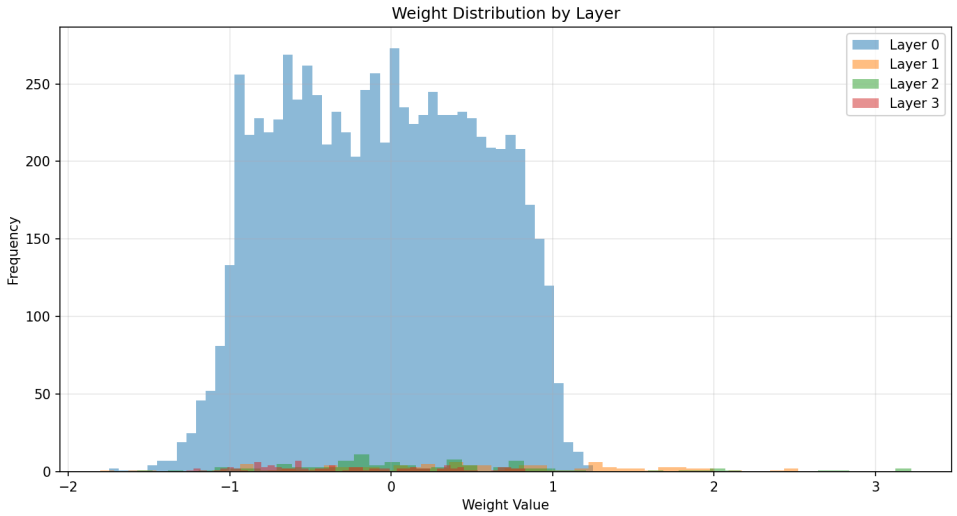
Softplus
Hasil prediksi - Test Accuracy: 0.1146 - `Test Accuracy: 0.1146`
Grafik loss



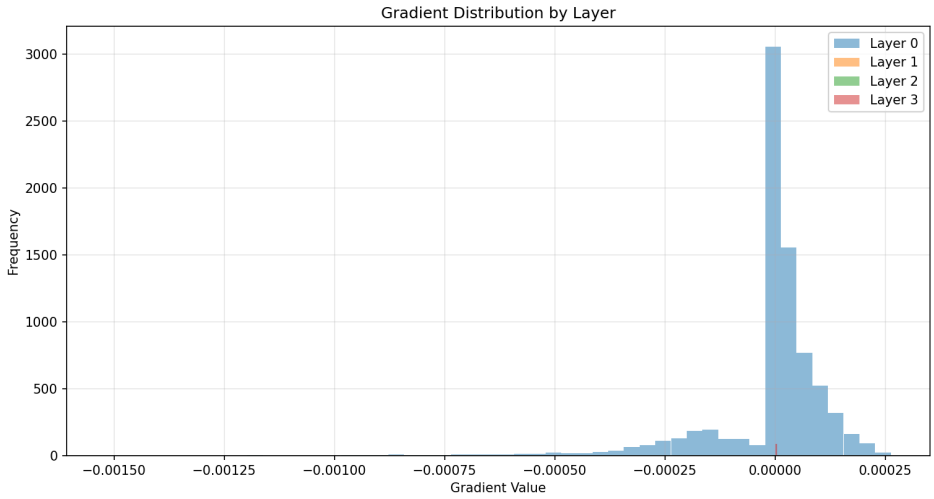
Iterasi Custom ANN:

- Epoch 1/10, Training Loss: 12.1418, Validation Loss: 12.0927
- Epoch 2/10, Training Loss: 13.0645, Validation Loss: 13.0138
- Epoch 3/10, Training Loss: 13.2916, Validation Loss: 13.2149
- Epoch 4/10, Training Loss: 15.2121, Validation Loss: 15.1369
- Epoch 5/10, Training Loss: 14.9071, Validation Loss: 14.7876
- Epoch 6/10, Training Loss: 13.0194, Validation Loss: 12.9913
- Epoch 7/10, Training Loss: 13.7016, Validation Loss: 13.7092
- Epoch 8/10, Training Loss: 14.1455, Validation Loss: 14.2041
- Epoch 9/10, Training Loss: 14.3573, Validation Loss: 14.3730
- Epoch 10/10, Training Loss: 14.1086, Validation Loss: 14.0678

Distribusi Bobot:



Distribusi Gradien Bobot:



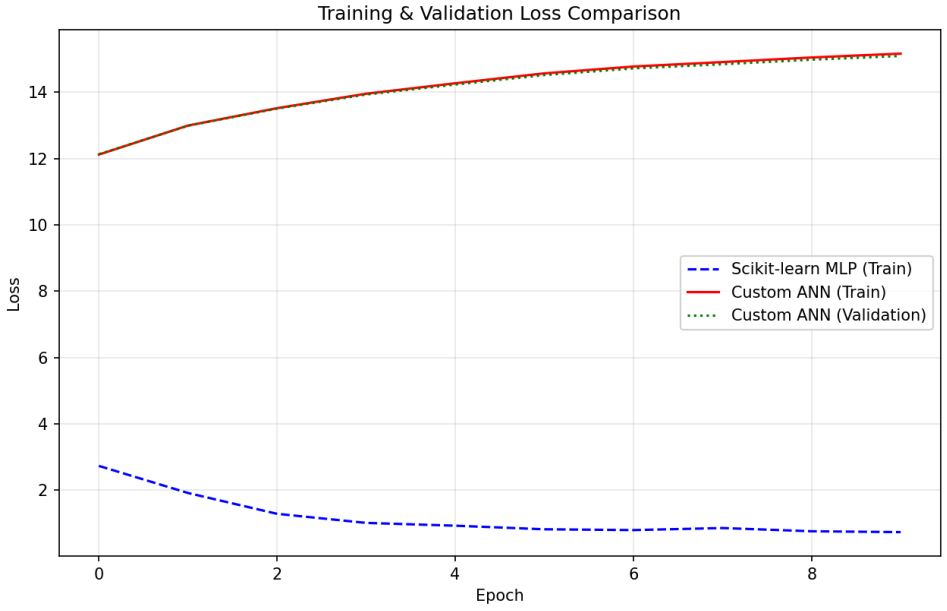
b. Leaky ReLU

Tanh

Hasil prediksi

- Test Accuracy: 0.1110
- `Test Accuracy: 0.1110`

Grafik loss

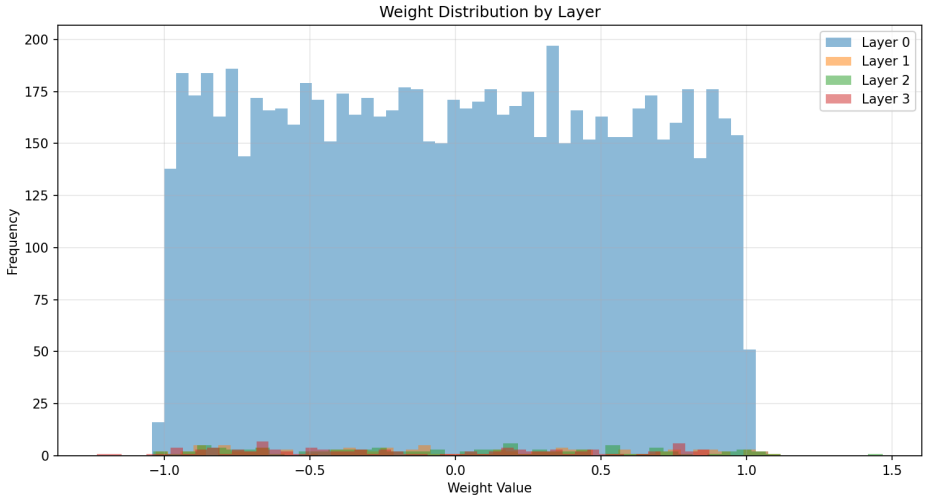


Iterasi Custom ANN:

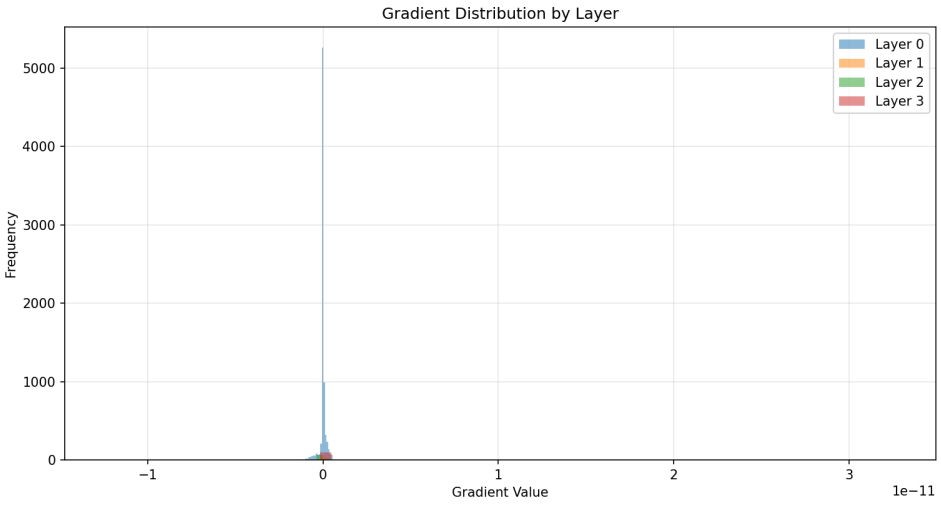
- Epoch 1/10, Training Loss: 12.1173, Validation Loss: 12.1269
- Epoch 2/10, Training Loss: 12.9907, Validation Loss: 12.9862

- Epoch 3/10, Training Loss: 13.5153, Validation Loss: 13.5023
- Epoch 4/10, Training Loss: 13.9514, Validation Loss: 13.9261
- Epoch 5/10, Training Loss: 14.2659, Validation Loss: 14.2271
- Epoch 6/10, Training Loss: 14.5645, Validation Loss: 14.5131
- Epoch 7/10, Training Loss: 14.7711, Validation Loss: 14.7137
- Epoch 8/10, Training Loss: 14.9051, Validation Loss: 14.8403
- Epoch 9/10, Training Loss: 15.0435, Validation Loss: 14.9767
- Epoch 10/10, Training Loss: 15.1598, Validation Loss: 15.0890

Distribusi Bobot:



Distribusi Gradien Bobot:



c. Mish

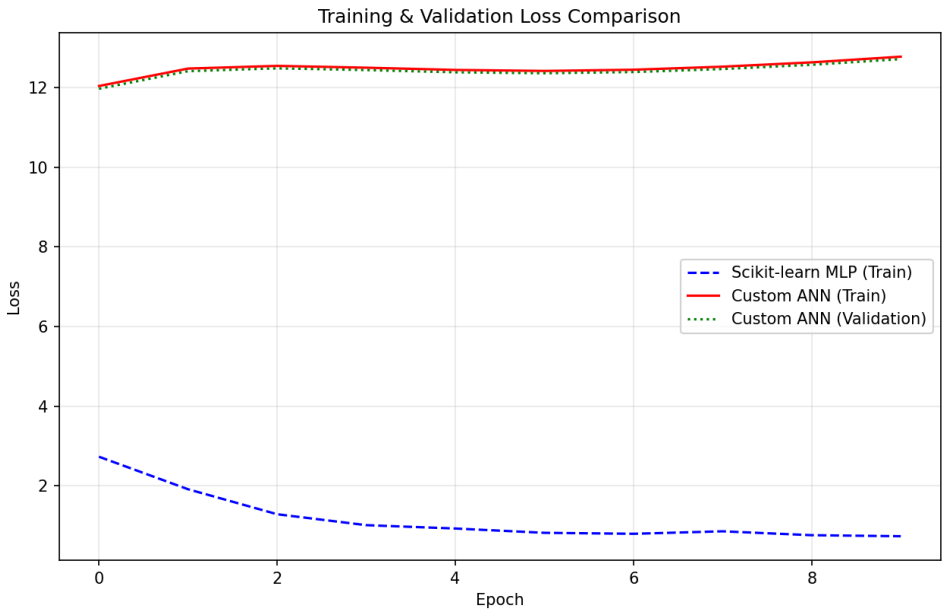
Tanh

Hasil prediksi

- Test Accuracy: 0.1646
-

```
Test Accuracy: 0.1646
```

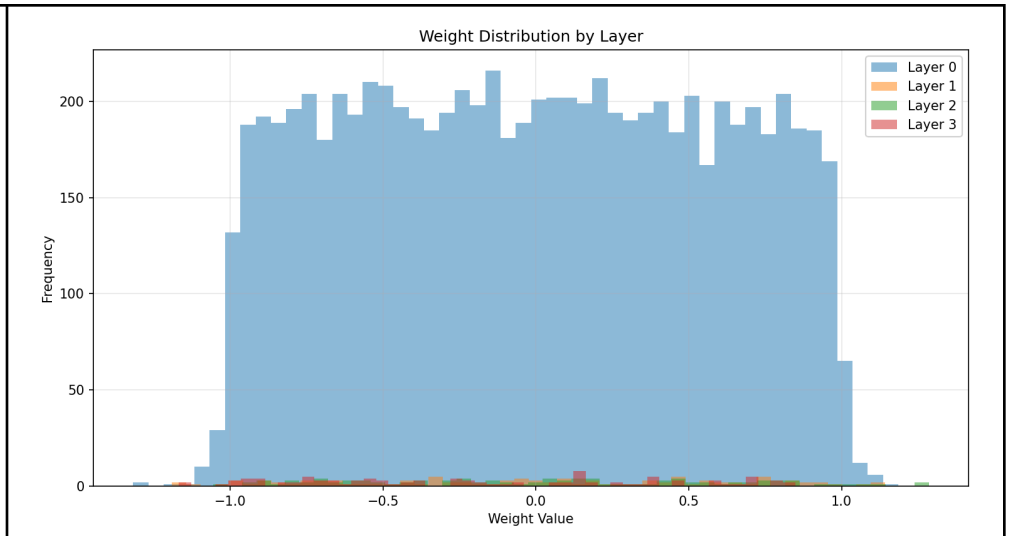
Grafik loss



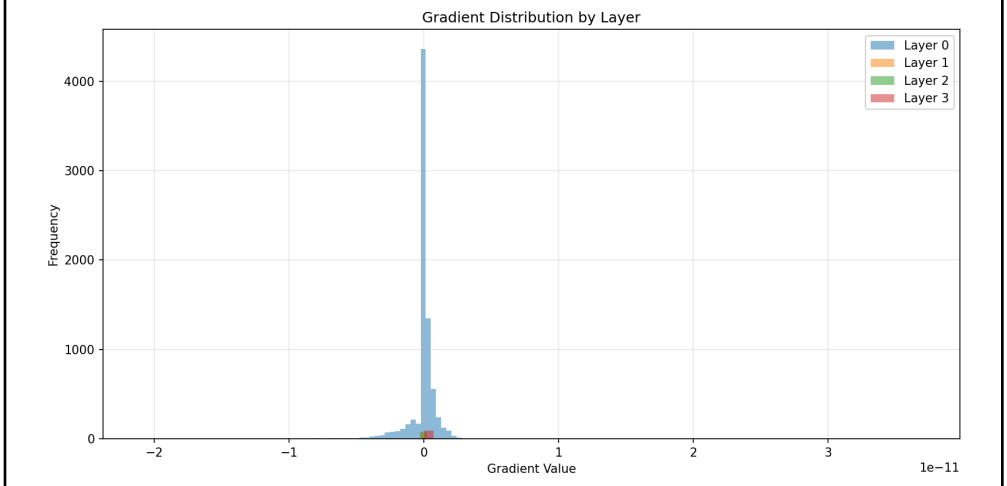
Iterasi Custom ANN:

- Epoch 1/10, Training Loss: 12.0413, Validation Loss: 11.9714
- Epoch 2/10, Training Loss: 12.4792, Validation Loss: 12.4136
- Epoch 3/10, Training Loss: 12.5460, Validation Loss: 12.4837
- Epoch 4/10, Training Loss: 12.5009, Validation Loss: 12.4409
- Epoch 5/10, Training Loss: 12.4446, Validation Loss: 12.3845
- Epoch 6/10, Training Loss: 12.4199, Validation Loss: 12.3610
- Epoch 7/10, Training Loss: 12.4498, Validation Loss: 12.3910
- Epoch 8/10, Training Loss: 12.5264, Validation Loss: 12.4661
- Epoch 9/10, Training Loss: 12.6346, Validation Loss: 12.5735
- Epoch 10/10, Training Loss: 12.7774, Validation Loss: 12.7161

Distribusi Bobot:



Distribusi Gradien Bobot:



d. Analisis

Dapat dilihat bahwa hanya fungsi Sigmoid yang menunjukkan kecenderungan penurunan loss selama iterasi pelatihan model. Model dengan fungsi aktivasi Sigmoid juga memiliki akurasi tertinggi diantara semua variasi. Dengan demikian, dapat disimpulkan bahwa fungsi aktivasi Sigmoid memiliki performa yang paling baik dalam pelatihan model.

3. Pengaruh **learning rate**

Hyperparameters	
Depth	3
Width semua layer	10

Loss	Categorical Cross Entropy
Batch Size	32
Epoch	50
Initialization	Xavier

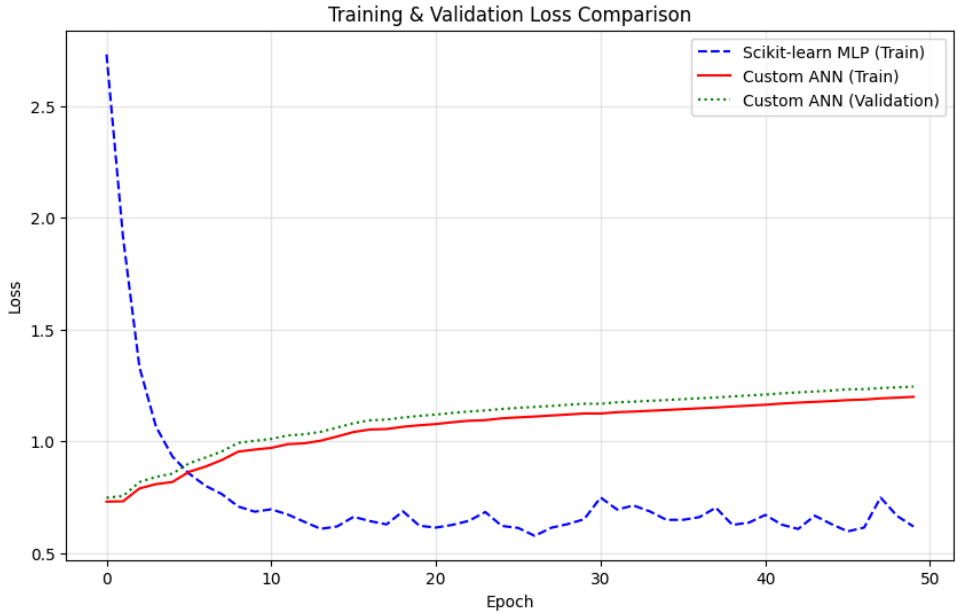
Learning rate = 0.1

Hasil prediksi

- Test Accuracy: 0.7759

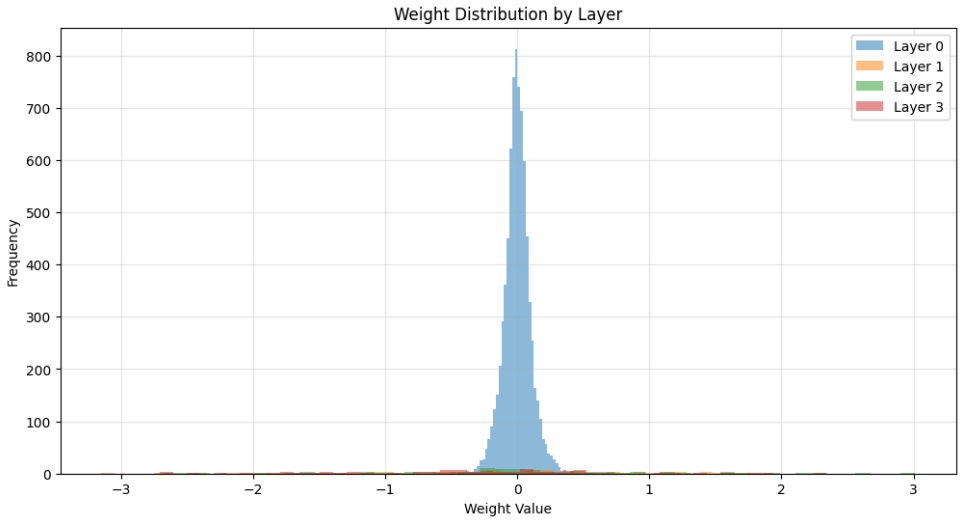
```
Test Accuracy: 0.7759
```

Grafik loss

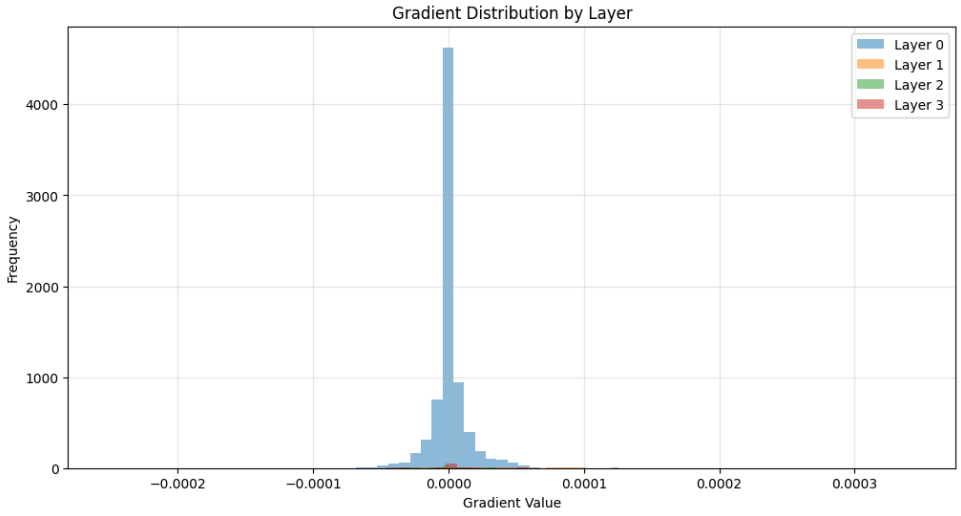


Iterasi Custom ANN:
Epoch 1/50, Training Loss: 0.7309, Validation Loss: 0.7486
Epoch 2/50, Training Loss: 0.7323, Validation Loss: 0.7557
Epoch 3/50, Training Loss: 0.7900, Validation Loss: 0.8183
Epoch 4/50, Training Loss: 0.8091, Validation Loss: 0.8415
Epoch 5/50, Training Loss: 0.8187, Validation Loss: 0.8562
Epoch 6/50, Training Loss: 0.8640, Validation Loss: 0.9018
Epoch 7/50, Training Loss: 0.8873, Validation Loss: 0.9281
Epoch 8/50, Training Loss: 0.9171, Validation Loss: 0.9554

	Epoch 9/50, Training Loss: 0.9546, Validation Loss: 0.9938 Epoch 10/50, Training Loss: 0.9638, Validation Loss: 1.0031 Epoch 11/50, Training Loss: 0.9712, Validation Loss: 1.0113 Epoch 12/50, Training Loss: 0.9879, Validation Loss: 1.0269 Epoch 13/50, Training Loss: 0.9917, Validation Loss: 1.0321 Epoch 14/50, Training Loss: 1.0035, Validation Loss: 1.0430 Epoch 15/50, Training Loss: 1.0222, Validation Loss: 1.0621 Epoch 16/50, Training Loss: 1.0422, Validation Loss: 1.0817 Epoch 17/50, Training Loss: 1.0536, Validation Loss: 1.0948 Epoch 18/50, Training Loss: 1.0556, Validation Loss: 1.0977 Epoch 19/50, Training Loss: 1.0656, Validation Loss: 1.1070 Epoch 20/50, Training Loss: 1.0725, Validation Loss: 1.1143 Epoch 21/50, Training Loss: 1.0778, Validation Loss: 1.1204 Epoch 22/50, Training Loss: 1.0861, Validation Loss: 1.1276 Epoch 23/50, Training Loss: 1.0925, Validation Loss: 1.1340 Epoch 24/50, Training Loss: 1.0953, Validation Loss: 1.1387 Epoch 25/50, Training Loss: 1.1034, Validation Loss: 1.1454 Epoch 26/50, Training Loss: 1.1078, Validation Loss: 1.1503 Epoch 27/50, Training Loss: 1.1115, Validation Loss: 1.1545 Epoch 28/50, Training Loss: 1.1161, Validation Loss: 1.1588 Epoch 29/50, Training Loss: 1.1204, Validation Loss: 1.1640 Epoch 30/50, Training Loss: 1.1251, Validation Loss: 1.1688 Epoch 31/50, Training Loss: 1.1250, Validation Loss: 1.1683 Epoch 32/50, Training Loss: 1.1309, Validation Loss: 1.1750 Epoch 33/50, Training Loss: 1.1338, Validation Loss: 1.1783 Epoch 34/50, Training Loss: 1.1373, Validation Loss: 1.1816 Epoch 35/50, Training Loss: 1.1408, Validation Loss: 1.1853 Epoch 36/50, Training Loss: 1.1446, Validation Loss: 1.1891 Epoch 37/50, Training Loss: 1.1483, Validation Loss: 1.1930 Epoch 38/50, Training Loss: 1.1519, Validation Loss: 1.1967 Epoch 39/50, Training Loss: 1.1562, Validation Loss: 1.2014 Epoch 40/50, Training Loss: 1.1603, Validation Loss: 1.2058 Epoch 41/50, Training Loss: 1.1641, Validation Loss: 1.2098 Epoch 42/50, Training Loss: 1.1695, Validation Loss: 1.2153 Epoch 43/50, Training Loss: 1.1737, Validation Loss: 1.2197 Epoch 44/50, Training Loss: 1.1772, Validation Loss: 1.2233 Epoch 45/50, Training Loss: 1.1806, Validation Loss: 1.2275 Epoch 46/50, Training Loss: 1.1850, Validation Loss: 1.2333 Epoch 47/50, Training Loss: 1.1874, Validation Loss: 1.2339 Epoch 48/50, Training Loss: 1.1927, Validation Loss: 1.2389 Epoch 49/50, Training Loss: 1.1960, Validation Loss: 1.2417 Epoch 50/50, Training Loss: 1.1996, Validation Loss: 1.2452
	Distribusi Bobot:



Distribusi Gradien Bobot:



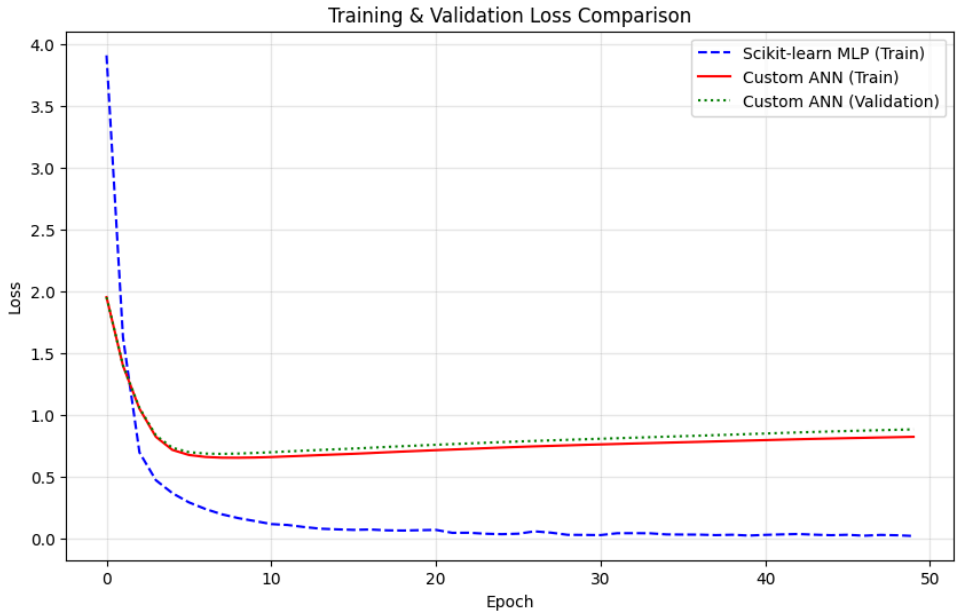
Learning rate = 0.001

Hasil prediksi

- Test Accuracy: 0.8331

```
Test Accuracy: 0.8331
```

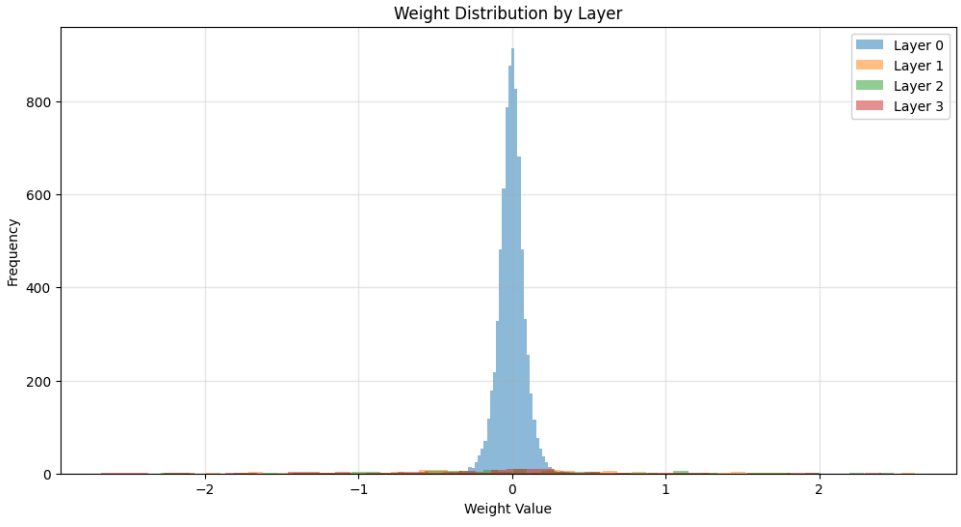
Grafik loss



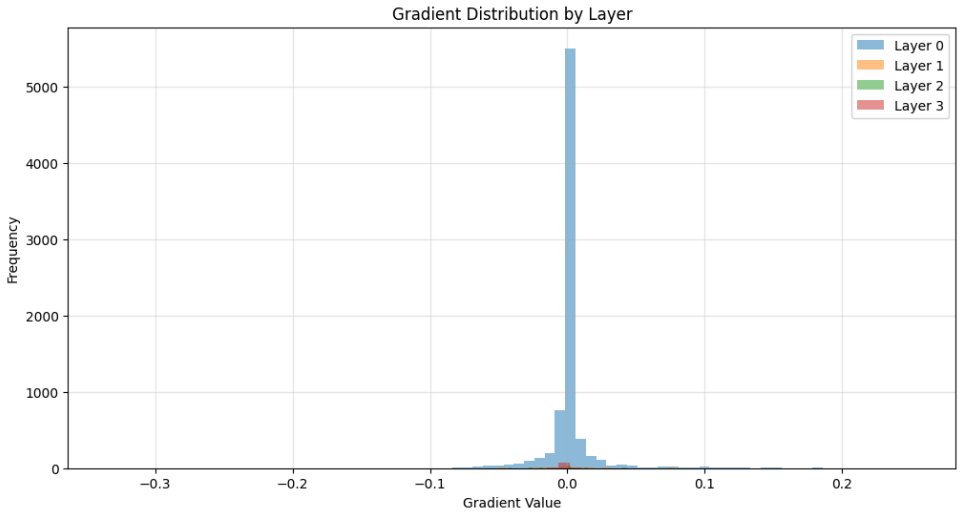
Iterasi Custom ANN:
Epoch 1/50, Training Loss: 1.9555, Validation Loss: 1.9548
Epoch 2/50, Training Loss: 1.4006, Validation Loss: 1.4033
Epoch 3/50, Training Loss: 1.0527, Validation Loss: 1.0591
Epoch 4/50, Training Loss: 0.8246, Validation Loss: 0.8368
Epoch 5/50, Training Loss: 0.7193, Validation Loss: 0.7365
Epoch 6/50, Training Loss: 0.6784, Validation Loss: 0.7013
Epoch 7/50, Training Loss: 0.6624, Validation Loss: 0.6896
Epoch 8/50, Training Loss: 0.6563, Validation Loss: 0.6871
Epoch 9/50, Training Loss: 0.6558, Validation Loss: 0.6899
Epoch 10/50, Training Loss: 0.6578, Validation Loss: 0.6946
Epoch 11/50, Training Loss: 0.6615, Validation Loss: 0.7003
Epoch 12/50, Training Loss: 0.6668, Validation Loss: 0.7071
Epoch 13/50, Training Loss: 0.6721, Validation Loss: 0.7132
Epoch 14/50, Training Loss: 0.6776, Validation Loss: 0.7192
Epoch 15/50, Training Loss: 0.6828, Validation Loss: 0.7248
Epoch 16/50, Training Loss: 0.6879, Validation Loss: 0.7304
Epoch 17/50, Training Loss: 0.6936, Validation Loss: 0.7365
Epoch 18/50, Training Loss: 0.7000, Validation Loss: 0.7432
Epoch 19/50, Training Loss: 0.7059, Validation Loss: 0.7494
Epoch 20/50, Training Loss: 0.7114, Validation Loss: 0.7550
Epoch 21/50, Training Loss: 0.7171, Validation Loss: 0.7608
Epoch 22/50, Training Loss: 0.7224, Validation Loss: 0.7662
Epoch 23/50, Training Loss: 0.7277, Validation Loss: 0.7716
Epoch 24/50, Training Loss: 0.7332, Validation Loss: 0.7772
Epoch 25/50, Training Loss: 0.7385, Validation Loss: 0.7825
Epoch 26/50, Training Loss: 0.7434, Validation Loss: 0.7874

Epoch 27/50, Training Loss: 0.7480, Validation Loss: 0.7922
Epoch 28/50, Training Loss: 0.7520, Validation Loss: 0.7965
Epoch 29/50, Training Loss: 0.7561, Validation Loss: 0.8010
Epoch 30/50, Training Loss: 0.7600, Validation Loss: 0.8054
Epoch 31/50, Training Loss: 0.7637, Validation Loss: 0.8096
Epoch 32/50, Training Loss: 0.7674, Validation Loss: 0.8139
Epoch 33/50, Training Loss: 0.7710, Validation Loss: 0.8181
Epoch 34/50, Training Loss: 0.7744, Validation Loss: 0.8222
Epoch 35/50, Training Loss: 0.7779, Validation Loss: 0.8264
Epoch 36/50, Training Loss: 0.7812, Validation Loss: 0.8304
Epoch 37/50, Training Loss: 0.7846, Validation Loss: 0.8345
Epoch 38/50, Training Loss: 0.7882, Validation Loss: 0.8388
Epoch 39/50, Training Loss: 0.7916, Validation Loss: 0.8431
Epoch 40/50, Training Loss: 0.7952, Validation Loss: 0.8475
Epoch 41/50, Training Loss: 0.7987, Validation Loss: 0.8520
Epoch 42/50, Training Loss: 0.8022, Validation Loss: 0.8564
Epoch 43/50, Training Loss: 0.8056, Validation Loss: 0.8608
Epoch 44/50, Training Loss: 0.8088, Validation Loss: 0.8649
Epoch 45/50, Training Loss: 0.8116, Validation Loss: 0.8686
Epoch 46/50, Training Loss: 0.8145, Validation Loss: 0.8722
Epoch 47/50, Training Loss: 0.8171, Validation Loss: 0.8756
Epoch 48/50, Training Loss: 0.8199, Validation Loss: 0.8790
Epoch 49/50, Training Loss: 0.8225, Validation Loss: 0.8823
Epoch 50/50, Training Loss: 0.8251, Validation Loss: 0.8854

Distribusi Bobot:



Distribusi Gradien Bobot:



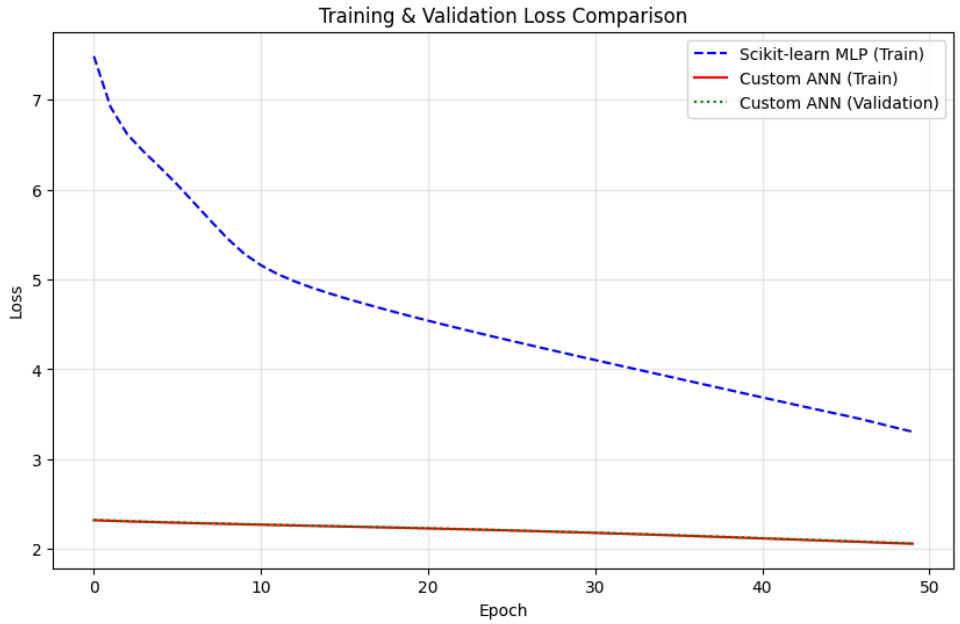
Learning rate = 0.00001

Hasil prediksi

- Test Accuracy: 0.3152

```
Test Accuracy: 0.3152
```

Grafik loss

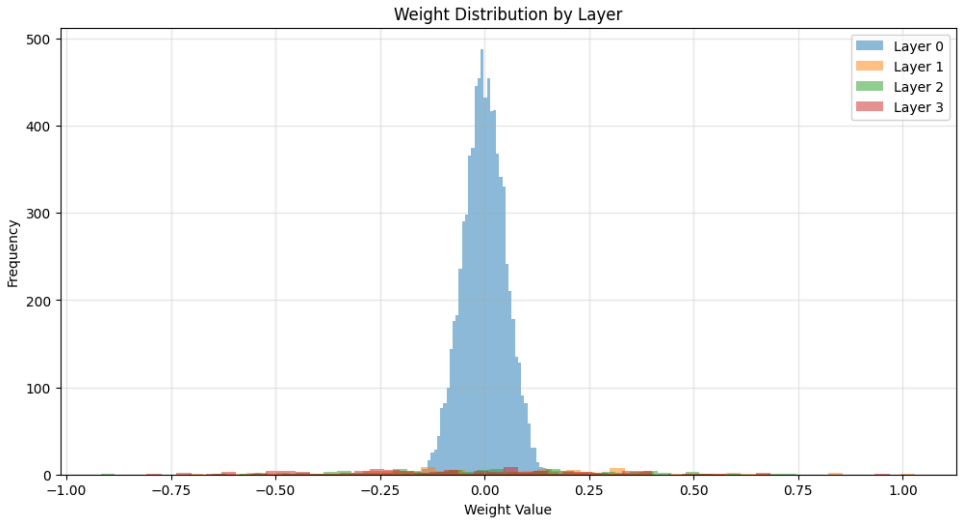


Iterasi Custom ANN:
Epoch 1/50, Training Loss: 2.3212, Validation Loss: 2.3247

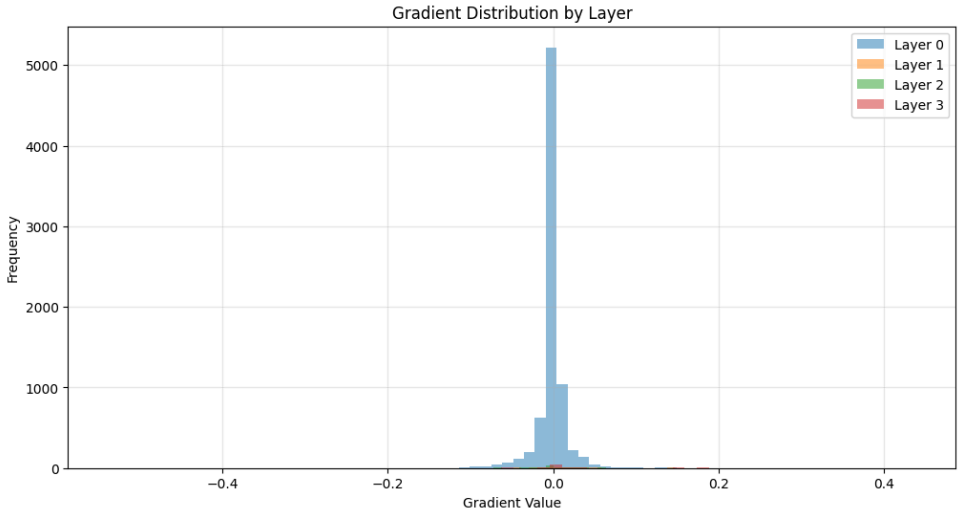
	Epoch 2/50, Training Loss: 2.3147, Validation Loss: 2.3181 Epoch 3/50, Training Loss: 2.3087, Validation Loss: 2.3120 Epoch 4/50, Training Loss: 2.3031, Validation Loss: 2.3063 Epoch 5/50, Training Loss: 2.2979, Validation Loss: 2.3009 Epoch 6/50, Training Loss: 2.2930, Validation Loss: 2.2959 Epoch 7/50, Training Loss: 2.2883, Validation Loss: 2.2912 Epoch 8/50, Training Loss: 2.2838, Validation Loss: 2.2866 Epoch 9/50, Training Loss: 2.2795, Validation Loss: 2.2822 Epoch 10/50, Training Loss: 2.2753, Validation Loss: 2.2779 Epoch 11/50, Training Loss: 2.2711, Validation Loss: 2.2737 Epoch 12/50, Training Loss: 2.2670, Validation Loss: 2.2695 Epoch 13/50, Training Loss: 2.2629, Validation Loss: 2.2654 Epoch 14/50, Training Loss: 2.2588, Validation Loss: 2.2613 Epoch 15/50, Training Loss: 2.2547, Validation Loss: 2.2571 Epoch 16/50, Training Loss: 2.2506, Validation Loss: 2.2530 Epoch 17/50, Training Loss: 2.2465, Validation Loss: 2.2489 Epoch 18/50, Training Loss: 2.2424, Validation Loss: 2.2448 Epoch 19/50, Training Loss: 2.2382, Validation Loss: 2.2406 Epoch 20/50, Training Loss: 2.2340, Validation Loss: 2.2363 Epoch 21/50, Training Loss: 2.2297, Validation Loss: 2.2320 Epoch 22/50, Training Loss: 2.2252, Validation Loss: 2.2276 Epoch 23/50, Training Loss: 2.2207, Validation Loss: 2.2231 Epoch 24/50, Training Loss: 2.2161, Validation Loss: 2.2184 Epoch 25/50, Training Loss: 2.2113, Validation Loss: 2.2137 Epoch 26/50, Training Loss: 2.2064, Validation Loss: 2.2088 Epoch 27/50, Training Loss: 2.2014, Validation Loss: 2.2039 Epoch 28/50, Training Loss: 2.1962, Validation Loss: 2.1987 Epoch 29/50, Training Loss: 2.1910, Validation Loss: 2.1935 Epoch 30/50, Training Loss: 2.1856, Validation Loss: 2.1881 Epoch 31/50, Training Loss: 2.1800, Validation Loss: 2.1826 Epoch 32/50, Training Loss: 2.1744, Validation Loss: 2.1770 Epoch 33/50, Training Loss: 2.1686, Validation Loss: 2.1713 Epoch 34/50, Training Loss: 2.1627, Validation Loss: 2.1655 Epoch 35/50, Training Loss: 2.1566, Validation Loss: 2.1595 Epoch 36/50, Training Loss: 2.1505, Validation Loss: 2.1534 Epoch 37/50, Training Loss: 2.1443, Validation Loss: 2.1473 Epoch 38/50, Training Loss: 2.1380, Validation Loss: 2.1411 Epoch 39/50, Training Loss: 2.1316, Validation Loss: 2.1348 Epoch 40/50, Training Loss: 2.1251, Validation Loss: 2.1284 Epoch 41/50, Training Loss: 2.1186, Validation Loss: 2.1220 Epoch 42/50, Training Loss: 2.1121, Validation Loss: 2.1156 Epoch 43/50, Training Loss: 2.1056, Validation Loss: 2.1092 Epoch 44/50, Training Loss: 2.0990, Validation Loss: 2.1027 Epoch 45/50, Training Loss: 2.0923, Validation Loss: 2.0962 Epoch 46/50, Training Loss: 2.0857, Validation Loss: 2.0896 Epoch 47/50, Training Loss: 2.0790, Validation Loss: 2.0831

Epoch 48/50, Training Loss: 2.0724, Validation Loss: 2.0765
Epoch 49/50, Training Loss: 2.0657, Validation Loss: 2.0699
Epoch 50/50, Training Loss: 2.0590, Validation Loss: 2.0634

Distribusi Bobot:



Distribusi Gradien Bobot:



Perbandingan

Dari percobaan yang dilakukan, dengan menggunakan learning rate 0,1 akurasi yang diperoleh adalah 0.7759, menggunakan learning rate 0,001, akurasi yang diperoleh adalah 0.8331, dan menggunakan learning rate 0,00001, akurasi yang diperoleh adalah 0.3152. Ini menunjukkan bahwa learning rate yang terlalu besar bisa membuat model tidak stabil sehingga loss naik dengan agresif dan menyebabkan akurasi rendah. Namun, jika learning rate yang digunakan terlalu rendah, model bisa terlalu lama untuk belajar sehingga menyebabkan

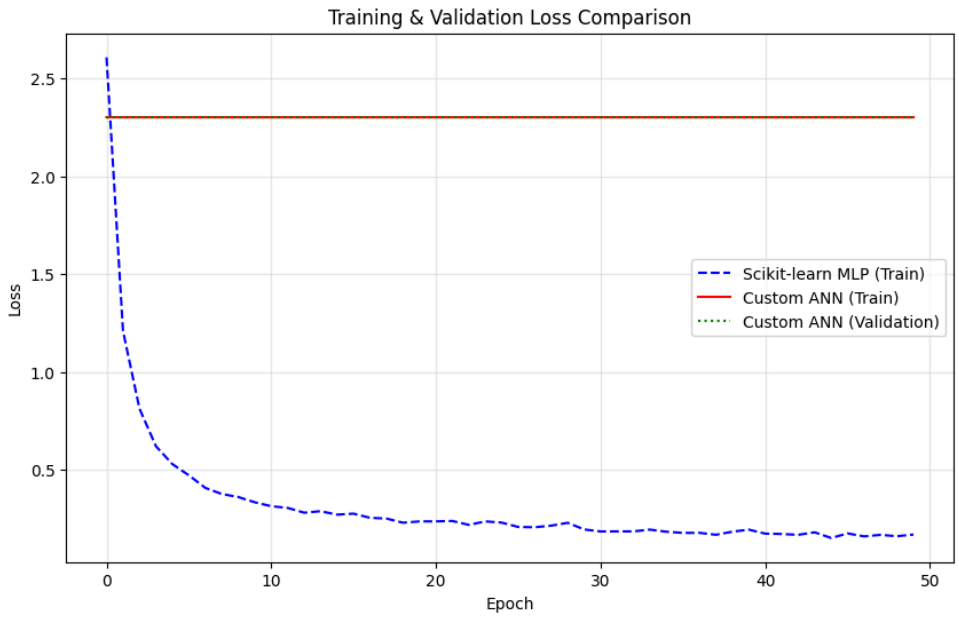
underfitting pada model. Oleh karena itu, diperlukan proses untuk memperoleh learning rate yang optimal supaya loss mengalami penurunan dengan stabil dan akurasi maksimal.

4. Pengaruh **inisialisasi bobot**

Hyperparameters	
Depth	3
Width semua layer	10
Loss	Categorical Cross Entropy
Batch Size	32
Epoch	50
Initialization	Xavier
Regularisasi	L2

a. Zero

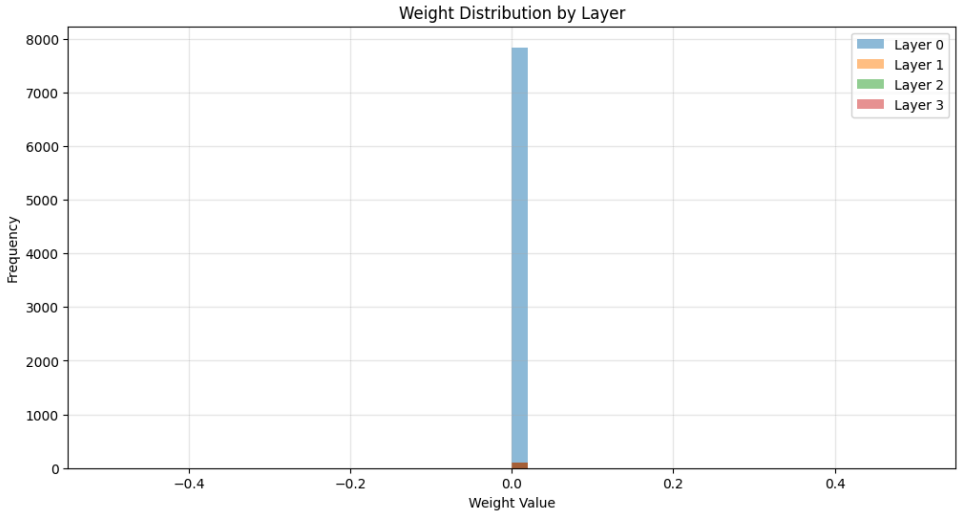
Zero	
	Hasil prediksi - Test Accuracy: 0.1126 `Test Accuracy: 0.1126`
	Grafik loss



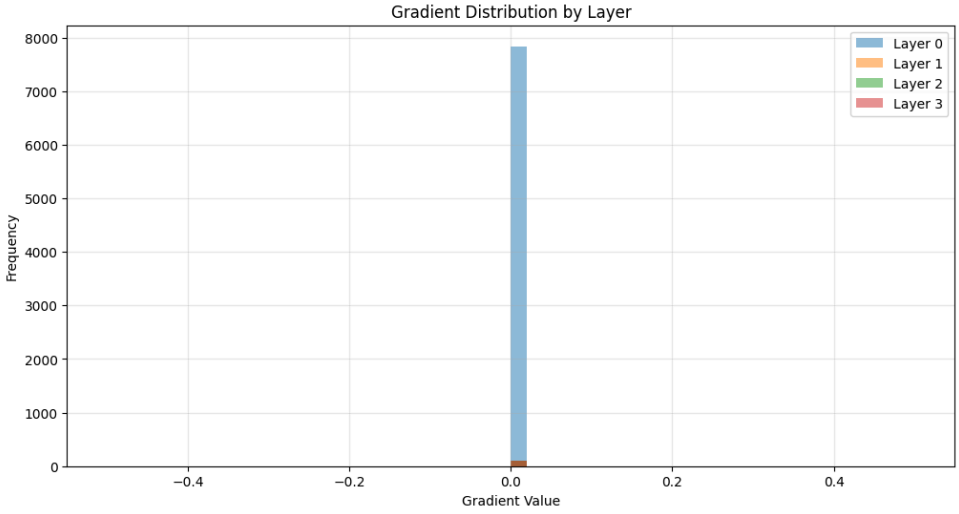
Iterasi Custom ANN:
Epoch 1/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 2/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 3/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 4/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 5/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 6/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 7/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 8/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 9/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 10/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 11/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 12/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 13/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 14/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 15/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 16/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 17/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 18/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 19/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 20/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 21/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 22/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 23/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 24/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 25/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 26/50, Training Loss: 2.3012, Validation Loss: 2.3012

Epoch 27/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 28/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 29/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 30/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 31/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 32/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 33/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 34/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 35/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 36/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 37/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 38/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 39/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 40/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 41/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 42/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 43/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 44/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 45/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 46/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 47/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 48/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 49/50, Training Loss: 2.3012, Validation Loss: 2.3012
Epoch 50/50, Training Loss: 2.3012, Validation Loss: 2.3012

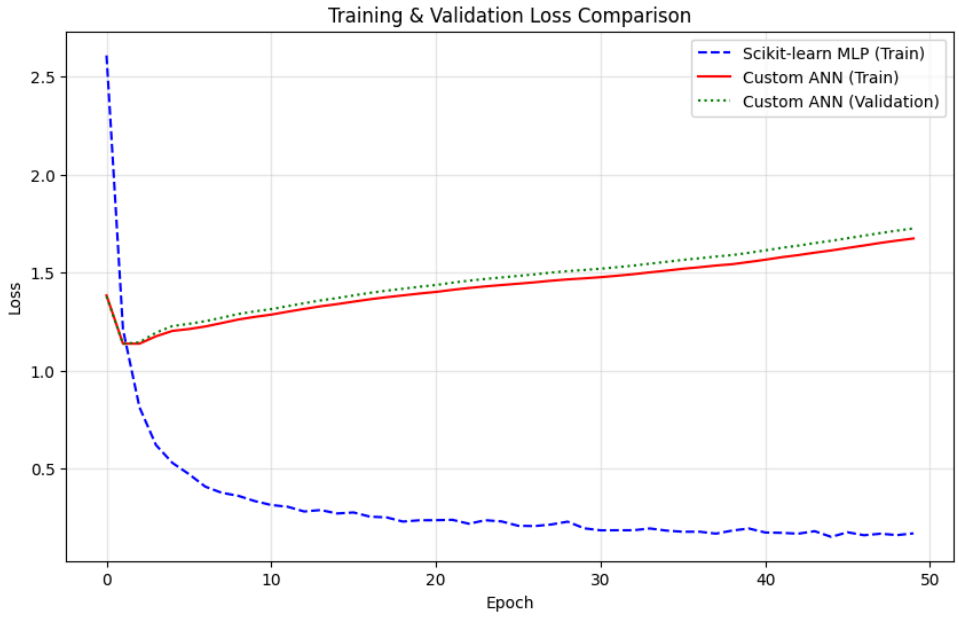
Distribusi Bobot:



Distribusi Gradien Bobot:



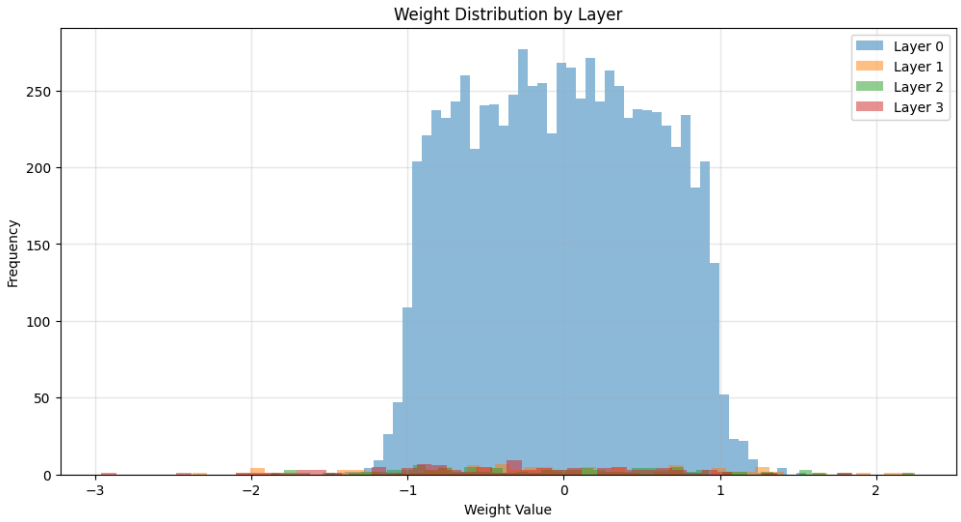
b. Uniform

Uniform
Hasil prediksi - Test Accuracy: 0.5945 `Test Accuracy: 0.5945`
Grafik loss 
Iterasi Custom ANN: Epoch 1/50, Training Loss: 1.3832, Validation Loss: 1.3773

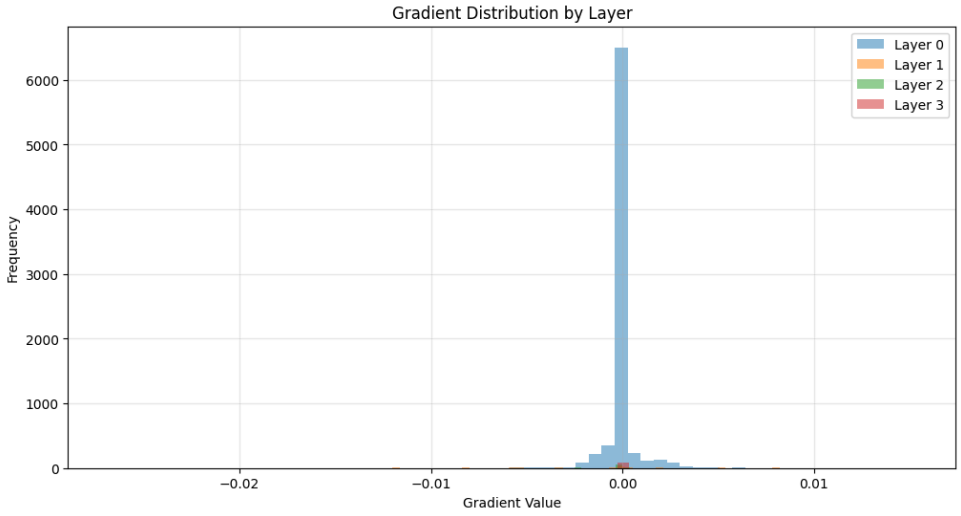
	Epoch 2/50, Training Loss: 1.1368, Validation Loss: 1.1359 Epoch 3/50, Training Loss: 1.1370, Validation Loss: 1.1440 Epoch 4/50, Training Loss: 1.1744, Validation Loss: 1.1929 Epoch 5/50, Training Loss: 1.2020, Validation Loss: 1.2266 Epoch 6/50, Training Loss: 1.2114, Validation Loss: 1.2379 Epoch 7/50, Training Loss: 1.2246, Validation Loss: 1.2518 Epoch 8/50, Training Loss: 1.2420, Validation Loss: 1.2693 Epoch 9/50, Training Loss: 1.2601, Validation Loss: 1.2881 Epoch 10/50, Training Loss: 1.2735, Validation Loss: 1.3017 Epoch 11/50, Training Loss: 1.2851, Validation Loss: 1.3132 Epoch 12/50, Training Loss: 1.3000, Validation Loss: 1.3282 Epoch 13/50, Training Loss: 1.3142, Validation Loss: 1.3433 Epoch 14/50, Training Loss: 1.3275, Validation Loss: 1.3575 Epoch 15/50, Training Loss: 1.3387, Validation Loss: 1.3695 Epoch 16/50, Training Loss: 1.3512, Validation Loss: 1.3830 Epoch 17/50, Training Loss: 1.3634, Validation Loss: 1.3961 Epoch 18/50, Training Loss: 1.3739, Validation Loss: 1.4073 Epoch 19/50, Training Loss: 1.3832, Validation Loss: 1.4172 Epoch 20/50, Training Loss: 1.3926, Validation Loss: 1.4273 Epoch 21/50, Training Loss: 1.4008, Validation Loss: 1.4362 Epoch 22/50, Training Loss: 1.4114, Validation Loss: 1.4479 Epoch 23/50, Training Loss: 1.4210, Validation Loss: 1.4582 Epoch 24/50, Training Loss: 1.4290, Validation Loss: 1.4671 Epoch 25/50, Training Loss: 1.4361, Validation Loss: 1.4749 Epoch 26/50, Training Loss: 1.4429, Validation Loss: 1.4825 Epoch 27/50, Training Loss: 1.4499, Validation Loss: 1.4905 Epoch 28/50, Training Loss: 1.4581, Validation Loss: 1.4997 Epoch 29/50, Training Loss: 1.4649, Validation Loss: 1.5072 Epoch 30/50, Training Loss: 1.4701, Validation Loss: 1.5128 Epoch 31/50, Training Loss: 1.4761, Validation Loss: 1.5193 Epoch 32/50, Training Loss: 1.4834, Validation Loss: 1.5270 Epoch 33/50, Training Loss: 1.4913, Validation Loss: 1.5353 Epoch 34/50, Training Loss: 1.5009, Validation Loss: 1.5452 Epoch 35/50, Training Loss: 1.5095, Validation Loss: 1.5544 Epoch 36/50, Training Loss: 1.5190, Validation Loss: 1.5645 Epoch 37/50, Training Loss: 1.5267, Validation Loss: 1.5724 Epoch 38/50, Training Loss: 1.5355, Validation Loss: 1.5815 Epoch 39/50, Training Loss: 1.5424, Validation Loss: 1.5889 Epoch 40/50, Training Loss: 1.5536, Validation Loss: 1.6005 Epoch 41/50, Training Loss: 1.5652, Validation Loss: 1.6128 Epoch 42/50, Training Loss: 1.5780, Validation Loss: 1.6260 Epoch 43/50, Training Loss: 1.5887, Validation Loss: 1.6370 Epoch 44/50, Training Loss: 1.6012, Validation Loss: 1.6500 Epoch 45/50, Training Loss: 1.6124, Validation Loss: 1.6616 Epoch 46/50, Training Loss: 1.6253, Validation Loss: 1.6750 Epoch 47/50, Training Loss: 1.6378, Validation Loss: 1.6878

Epoch 48/50, Training Loss: 1.6511, Validation Loss: 1.7015
Epoch 49/50, Training Loss: 1.6628, Validation Loss: 1.7136
Epoch 50/50, Training Loss: 1.6736, Validation Loss: 1.7247

Distribusi Bobot:



Distribusi Gradien Bobot:



c. Normal

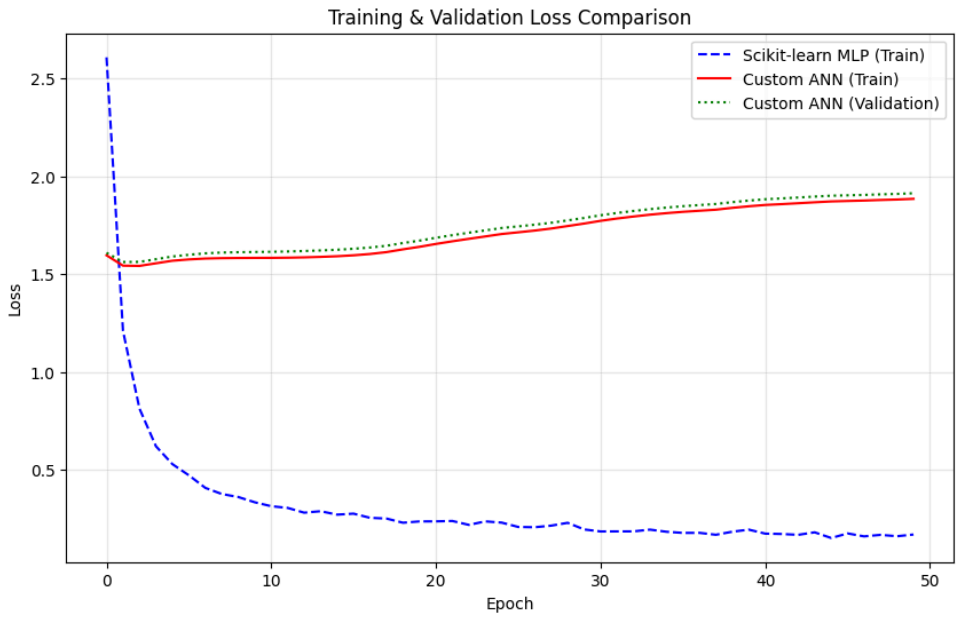
Normal

Hasil prediksi

- Test Accuracy: 0.4908

```
Test Accuracy: 0.4908
```

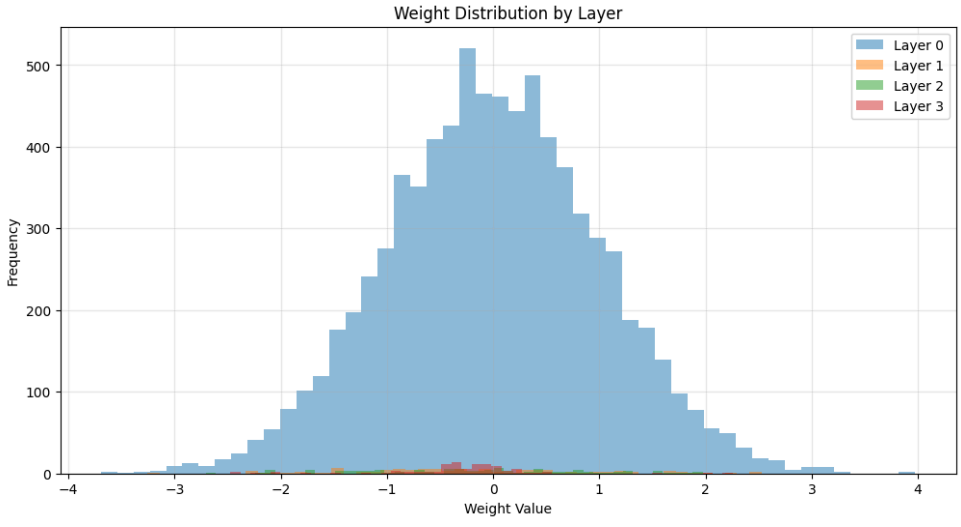
Grafik loss



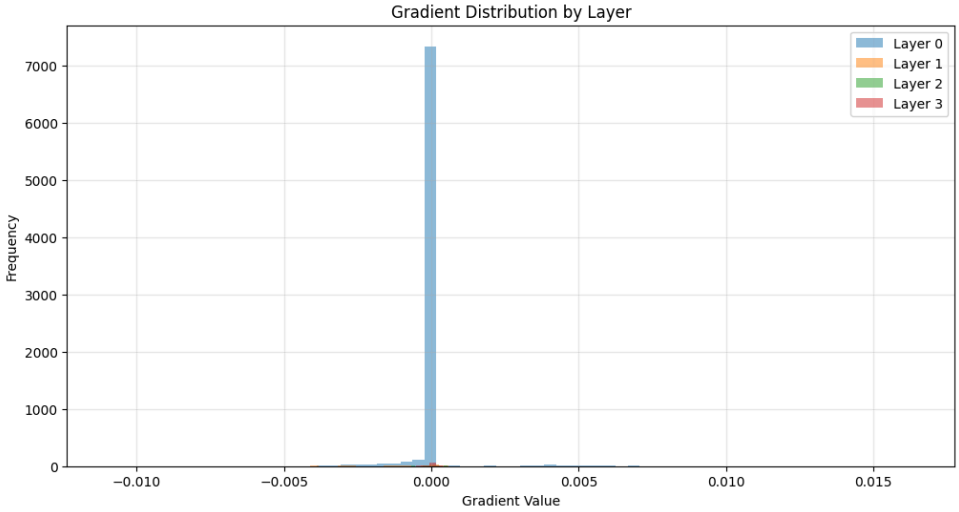
Iterasi Custom ANN:
Epoch 1/50, Training Loss: 1.5959, Validation Loss: 1.6086
Epoch 2/50, Training Loss: 1.5429, Validation Loss: 1.5616
Epoch 3/50, Training Loss: 1.5417, Validation Loss: 1.5623
Epoch 4/50, Training Loss: 1.5549, Validation Loss: 1.5759
Epoch 5/50, Training Loss: 1.5681, Validation Loss: 1.5898
Epoch 6/50, Training Loss: 1.5748, Validation Loss: 1.5985
Epoch 7/50, Training Loss: 1.5793, Validation Loss: 1.6060
Epoch 8/50, Training Loss: 1.5813, Validation Loss: 1.6099
Epoch 9/50, Training Loss: 1.5822, Validation Loss: 1.6117
Epoch 10/50, Training Loss: 1.5826, Validation Loss: 1.6129
Epoch 11/50, Training Loss: 1.5828, Validation Loss: 1.6138
Epoch 12/50, Training Loss: 1.5837, Validation Loss: 1.6153
Epoch 13/50, Training Loss: 1.5853, Validation Loss: 1.6177
Epoch 14/50, Training Loss: 1.5877, Validation Loss: 1.6206
Epoch 15/50, Training Loss: 1.5910, Validation Loss: 1.6243
Epoch 16/50, Training Loss: 1.5958, Validation Loss: 1.6292
Epoch 17/50, Training Loss: 1.6025, Validation Loss: 1.6357
Epoch 18/50, Training Loss: 1.6122, Validation Loss: 1.6446
Epoch 19/50, Training Loss: 1.6263, Validation Loss: 1.6585
Epoch 20/50, Training Loss: 1.6391, Validation Loss: 1.6707
Epoch 21/50, Training Loss: 1.6539, Validation Loss: 1.6855
Epoch 22/50, Training Loss: 1.6674, Validation Loss: 1.6986
Epoch 23/50, Training Loss: 1.6800, Validation Loss: 1.7113
Epoch 24/50, Training Loss: 1.6922, Validation Loss: 1.7233
Epoch 25/50, Training Loss: 1.7047, Validation Loss: 1.7355
Epoch 26/50, Training Loss: 1.7133, Validation Loss: 1.7437

Epoch 27/50, Training Loss: 1.7222, Validation Loss: 1.7522
Epoch 28/50, Training Loss: 1.7327, Validation Loss: 1.7621
Epoch 29/50, Training Loss: 1.7453, Validation Loss: 1.7743
Epoch 30/50, Training Loss: 1.7580, Validation Loss: 1.7866
Epoch 31/50, Training Loss: 1.7720, Validation Loss: 1.8005
Epoch 32/50, Training Loss: 1.7838, Validation Loss: 1.8124
Epoch 33/50, Training Loss: 1.7940, Validation Loss: 1.8224
Epoch 34/50, Training Loss: 1.8035, Validation Loss: 1.8317
Epoch 35/50, Training Loss: 1.8113, Validation Loss: 1.8398
Epoch 36/50, Training Loss: 1.8180, Validation Loss: 1.8468
Epoch 37/50, Training Loss: 1.8235, Validation Loss: 1.8526
Epoch 38/50, Training Loss: 1.8289, Validation Loss: 1.8581
Epoch 39/50, Training Loss: 1.8383, Validation Loss: 1.8675
Epoch 40/50, Training Loss: 1.8463, Validation Loss: 1.8755
Epoch 41/50, Training Loss: 1.8531, Validation Loss: 1.8824
Epoch 42/50, Training Loss: 1.8573, Validation Loss: 1.8866
Epoch 43/50, Training Loss: 1.8620, Validation Loss: 1.8912
Epoch 44/50, Training Loss: 1.8667, Validation Loss: 1.8960
Epoch 45/50, Training Loss: 1.8711, Validation Loss: 1.9003
Epoch 46/50, Training Loss: 1.8734, Validation Loss: 1.9024
Epoch 47/50, Training Loss: 1.8756, Validation Loss: 1.9044
Epoch 48/50, Training Loss: 1.8786, Validation Loss: 1.9073
Epoch 49/50, Training Loss: 1.8809, Validation Loss: 1.9096
Epoch 50/50, Training Loss: 1.8844, Validation Loss: 1.9129

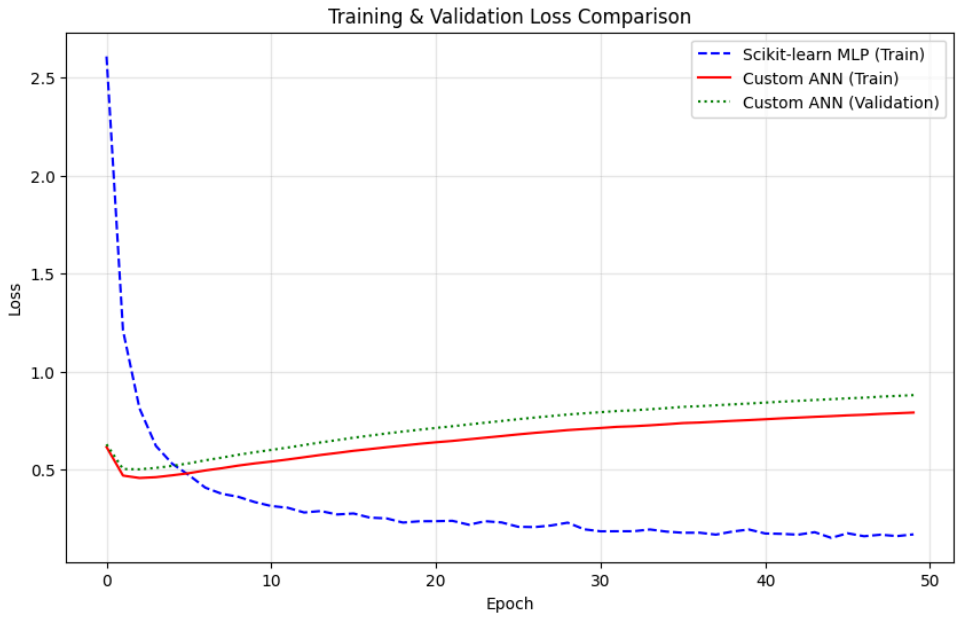
Distribusi Bobot:



Distribusi Gradien Bobot:



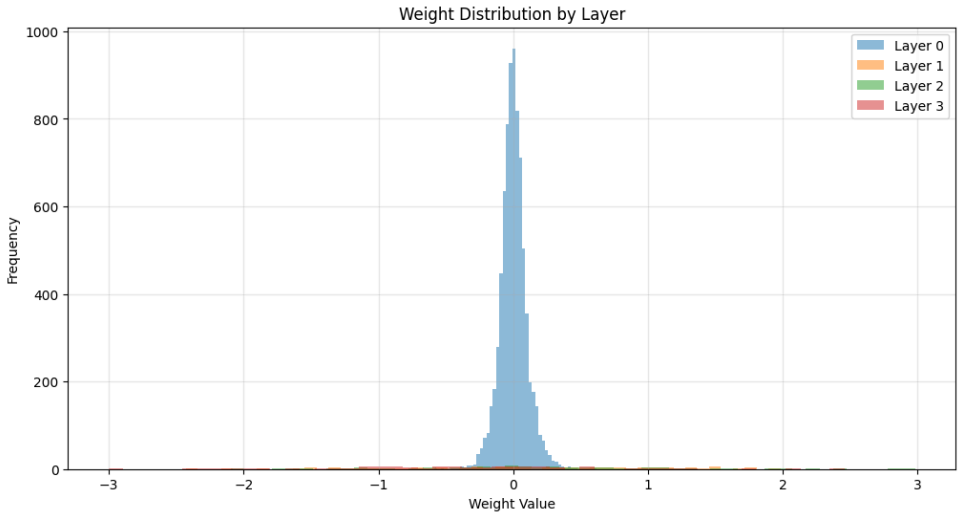
d. Xavier

Xavier
Hasil prediksi - Test Accuracy: 0.8496 `Test Accuracy: 0.8496`
Grafik loss 
Iterasi Custom ANN: Epoch 1/50, Training Loss: 0.6148, Validation Loss: 0.6284 Epoch 2/50, Training Loss: 0.4689, Validation Loss: 0.5025

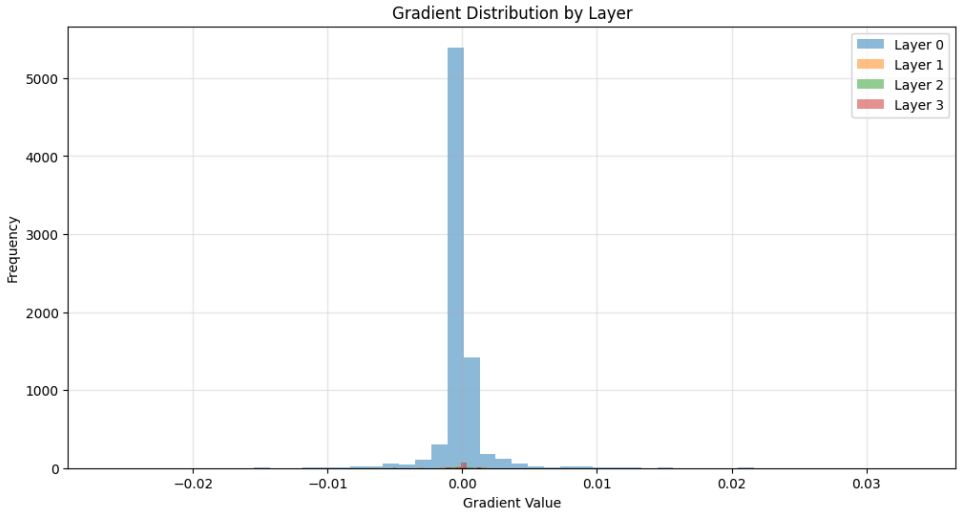
	Epoch 3/50, Training Loss: 0.4567, Validation Loss: 0.5009 Epoch 4/50, Training Loss: 0.4608, Validation Loss: 0.5081 Epoch 5/50, Training Loss: 0.4704, Validation Loss: 0.5186 Epoch 6/50, Training Loss: 0.4811, Validation Loss: 0.5314 Epoch 7/50, Training Loss: 0.4951, Validation Loss: 0.5471 Epoch 8/50, Training Loss: 0.5066, Validation Loss: 0.5605 Epoch 9/50, Training Loss: 0.5199, Validation Loss: 0.5751 Epoch 10/50, Training Loss: 0.5310, Validation Loss: 0.5882 Epoch 11/50, Training Loss: 0.5406, Validation Loss: 0.5997 Epoch 12/50, Training Loss: 0.5511, Validation Loss: 0.6119 Epoch 13/50, Training Loss: 0.5626, Validation Loss: 0.6248 Epoch 14/50, Training Loss: 0.5742, Validation Loss: 0.6381 Epoch 15/50, Training Loss: 0.5844, Validation Loss: 0.6498 Epoch 16/50, Training Loss: 0.5953, Validation Loss: 0.6623 Epoch 17/50, Training Loss: 0.6042, Validation Loss: 0.6729 Epoch 18/50, Training Loss: 0.6135, Validation Loss: 0.6838 Epoch 19/50, Training Loss: 0.6222, Validation Loss: 0.6934 Epoch 20/50, Training Loss: 0.6310, Validation Loss: 0.7031 Epoch 21/50, Training Loss: 0.6389, Validation Loss: 0.7121 Epoch 22/50, Training Loss: 0.6457, Validation Loss: 0.7205 Epoch 23/50, Training Loss: 0.6541, Validation Loss: 0.7297 Epoch 24/50, Training Loss: 0.6624, Validation Loss: 0.7389 Epoch 25/50, Training Loss: 0.6706, Validation Loss: 0.7477 Epoch 26/50, Training Loss: 0.6791, Validation Loss: 0.7567 Epoch 27/50, Training Loss: 0.6868, Validation Loss: 0.7649 Epoch 28/50, Training Loss: 0.6936, Validation Loss: 0.7726 Epoch 29/50, Training Loss: 0.7010, Validation Loss: 0.7804 Epoch 30/50, Training Loss: 0.7065, Validation Loss: 0.7866 Epoch 31/50, Training Loss: 0.7118, Validation Loss: 0.7924 Epoch 32/50, Training Loss: 0.7176, Validation Loss: 0.7984 Epoch 33/50, Training Loss: 0.7206, Validation Loss: 0.8019 Epoch 34/50, Training Loss: 0.7254, Validation Loss: 0.8073 Epoch 35/50, Training Loss: 0.7308, Validation Loss: 0.8130 Epoch 36/50, Training Loss: 0.7368, Validation Loss: 0.8197 Epoch 37/50, Training Loss: 0.7397, Validation Loss: 0.8229 Epoch 38/50, Training Loss: 0.7440, Validation Loss: 0.8278 Epoch 39/50, Training Loss: 0.7482, Validation Loss: 0.8321 Epoch 40/50, Training Loss: 0.7520, Validation Loss: 0.8365 Epoch 41/50, Training Loss: 0.7564, Validation Loss: 0.8414 Epoch 42/50, Training Loss: 0.7610, Validation Loss: 0.8460 Epoch 43/50, Training Loss: 0.7647, Validation Loss: 0.8501 Epoch 44/50, Training Loss: 0.7686, Validation Loss: 0.8546 Epoch 45/50, Training Loss: 0.7722, Validation Loss: 0.8587 Epoch 46/50, Training Loss: 0.7762, Validation Loss: 0.8632 Epoch 47/50, Training Loss: 0.7793, Validation Loss: 0.8664 Epoch 48/50, Training Loss: 0.7837, Validation Loss: 0.8717

Epoch 49/50, Training Loss: 0.7871, Validation Loss: 0.8754
Epoch 50/50, Training Loss: 0.7905, Validation Loss: 0.8794
Model saved successfully to saved_models/my_ann_model.pkl

Distribusi Bobot:



Distribusi Gradien Bobot:



e. He

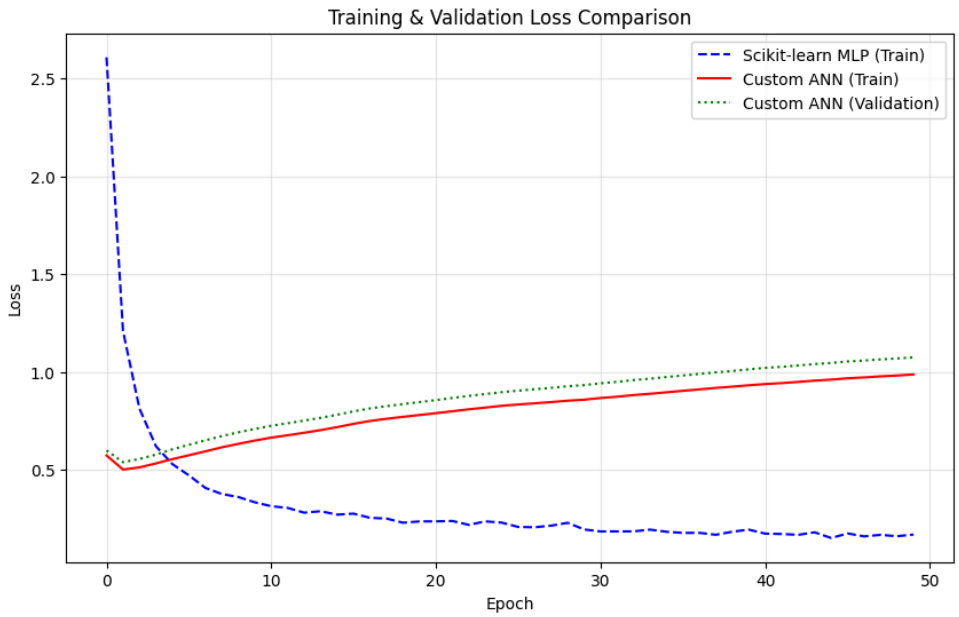
He

Hasil prediksi

- Test Accuracy: 0.8250

```
Test Accuracy: 0.8250
```

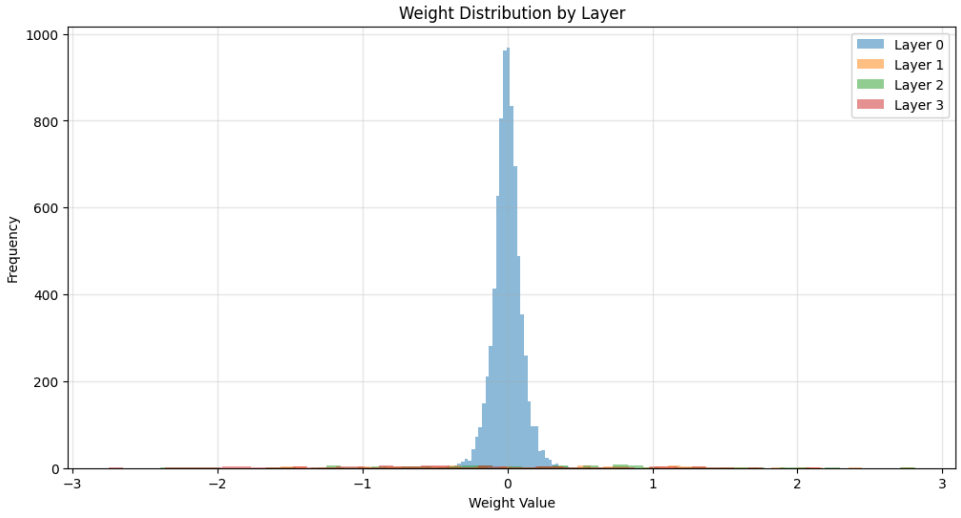
Grafik loss



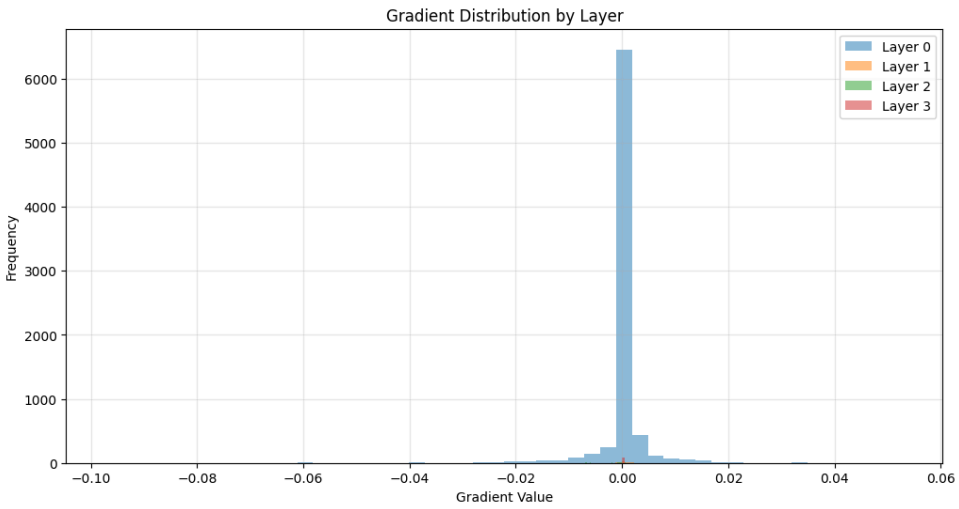
Iterasi Custom ANN:
Epoch 1/50, Training Loss: 0.5730, Validation Loss: 0.5986
Epoch 2/50, Training Loss: 0.5009, Validation Loss: 0.5386
Epoch 3/50, Training Loss: 0.5121, Validation Loss: 0.5552
Epoch 4/50, Training Loss: 0.5324, Validation Loss: 0.5772
Epoch 5/50, Training Loss: 0.5543, Validation Loss: 0.6043
Epoch 6/50, Training Loss: 0.5739, Validation Loss: 0.6275
Epoch 7/50, Training Loss: 0.5940, Validation Loss: 0.6502
Epoch 8/50, Training Loss: 0.6147, Validation Loss: 0.6720
Epoch 9/50, Training Loss: 0.6322, Validation Loss: 0.6912
Epoch 10/50, Training Loss: 0.6491, Validation Loss: 0.7082
Epoch 11/50, Training Loss: 0.6641, Validation Loss: 0.7246
Epoch 12/50, Training Loss: 0.6761, Validation Loss: 0.7376
Epoch 13/50, Training Loss: 0.6890, Validation Loss: 0.7513
Epoch 14/50, Training Loss: 0.7025, Validation Loss: 0.7651
Epoch 15/50, Training Loss: 0.7176, Validation Loss: 0.7809
Epoch 16/50, Training Loss: 0.7346, Validation Loss: 0.7983
Epoch 17/50, Training Loss: 0.7492, Validation Loss: 0.8133
Epoch 18/50, Training Loss: 0.7604, Validation Loss: 0.8249
Epoch 19/50, Training Loss: 0.7702, Validation Loss: 0.8353
Epoch 20/50, Training Loss: 0.7797, Validation Loss: 0.8456
Epoch 21/50, Training Loss: 0.7889, Validation Loss: 0.8558
Epoch 22/50, Training Loss: 0.7988, Validation Loss: 0.8668
Epoch 23/50, Training Loss: 0.8088, Validation Loss: 0.8773
Epoch 24/50, Training Loss: 0.8175, Validation Loss: 0.8868
Epoch 25/50, Training Loss: 0.8267, Validation Loss: 0.8965
Epoch 26/50, Training Loss: 0.8334, Validation Loss: 0.9042

Epoch 27/50, Training Loss: 0.8395, Validation Loss: 0.9115
Epoch 28/50, Training Loss: 0.8457, Validation Loss: 0.9185
Epoch 29/50, Training Loss: 0.8528, Validation Loss: 0.9267
Epoch 30/50, Training Loss: 0.8580, Validation Loss: 0.9325
Epoch 31/50, Training Loss: 0.8666, Validation Loss: 0.9418
Epoch 32/50, Training Loss: 0.8731, Validation Loss: 0.9490
Epoch 33/50, Training Loss: 0.8816, Validation Loss: 0.9582
Epoch 34/50, Training Loss: 0.8882, Validation Loss: 0.9653
Epoch 35/50, Training Loss: 0.8958, Validation Loss: 0.9737
Epoch 36/50, Training Loss: 0.9033, Validation Loss: 0.9817
Epoch 37/50, Training Loss: 0.9107, Validation Loss: 0.9898
Epoch 38/50, Training Loss: 0.9182, Validation Loss: 0.9977
Epoch 39/50, Training Loss: 0.9246, Validation Loss: 1.0047
Epoch 40/50, Training Loss: 0.9317, Validation Loss: 1.0130
Epoch 41/50, Training Loss: 0.9379, Validation Loss: 1.0205
Epoch 42/50, Training Loss: 0.9427, Validation Loss: 1.0261
Epoch 43/50, Training Loss: 0.9489, Validation Loss: 1.0327
Epoch 44/50, Training Loss: 0.9553, Validation Loss: 1.0400
Epoch 45/50, Training Loss: 0.9604, Validation Loss: 1.0457
Epoch 46/50, Training Loss: 0.9674, Validation Loss: 1.0531
Epoch 47/50, Training Loss: 0.9717, Validation Loss: 1.0579
Epoch 48/50, Training Loss: 0.9772, Validation Loss: 1.0639
Epoch 49/50, Training Loss: 0.9814, Validation Loss: 1.0686
Epoch 50/50, Training Loss: 0.9870, Validation Loss: 1.0744

Distribusi Bobot:



Distribusi Gradien Bobot:



Perbandingan

Inisiasi menggunakan Xavier dan He sangat unggul karena bisa mempertahankan variansi dari gradien dengan maksimal. Inisiasi menggunakan Zero mengalami kegagalan karena tidak terdapat pembelajaran. Ini dibuktikan dengan gradien yang bernilai 0. Inisiasi menggunakan uniform dan normal kurang optimal untuk arsitektur FFNN ini.

5. Pengaruh **regularisasi**

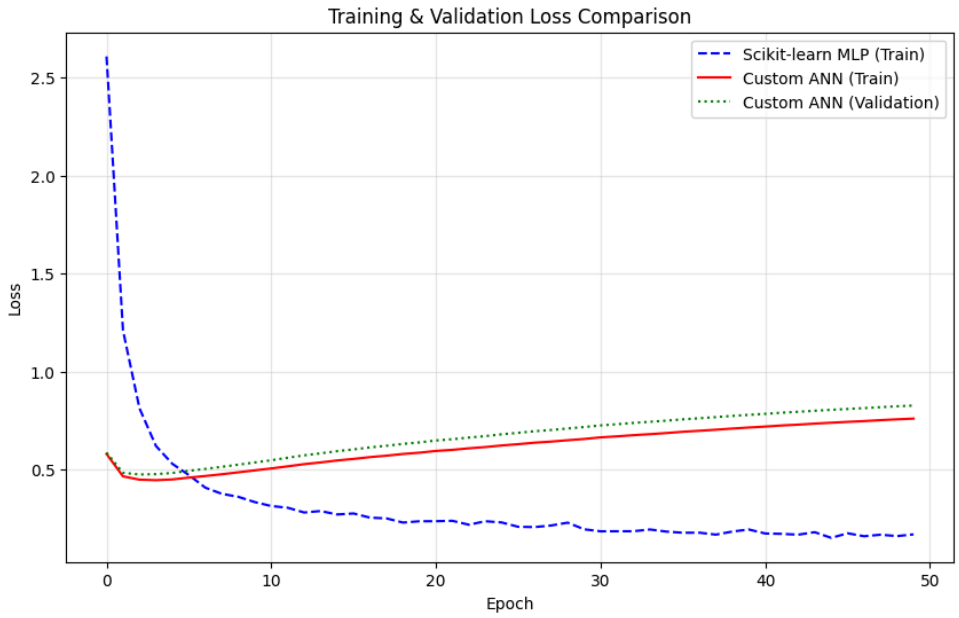
Hyperparameters	
Depth	3
Width semua layer	10
Loss	Categorical Cross Entropy
Batch Size	32
Epoch	50
Initialization	Xavier

a. Tanpa regularisasi

Hasil prediksi
- Test Accuracy: 0.8540

```
Test Accuracy: 0.8540
```

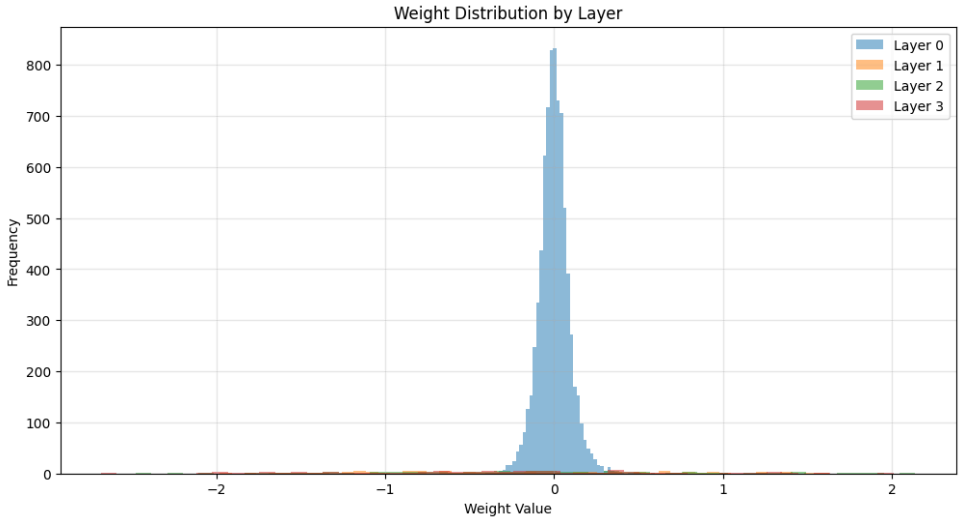
Grafik loss



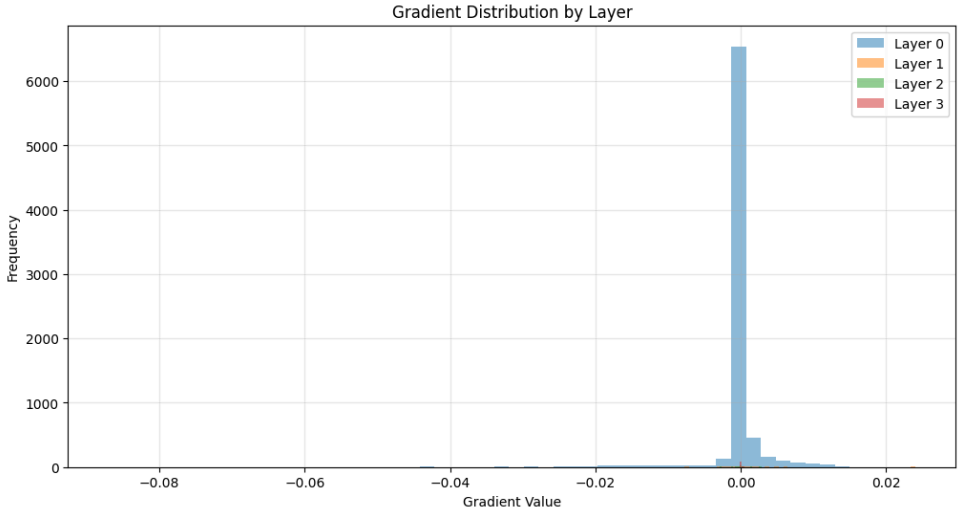
Iterasi Custom ANN:
Epoch 1/50, Training Loss: 0.5793, Validation Loss: 0.5857
Epoch 2/50, Training Loss: 0.4650, Validation Loss: 0.4830
Epoch 3/50, Training Loss: 0.4482, Validation Loss: 0.4745
Epoch 4/50, Training Loss: 0.4454, Validation Loss: 0.4764
Epoch 5/50, Training Loss: 0.4492, Validation Loss: 0.4822
Epoch 6/50, Training Loss: 0.4586, Validation Loss: 0.4933
Epoch 7/50, Training Loss: 0.4668, Validation Loss: 0.5032
Epoch 8/50, Training Loss: 0.4757, Validation Loss: 0.5137
Epoch 9/50, Training Loss: 0.4855, Validation Loss: 0.5245
Epoch 10/50, Training Loss: 0.4955, Validation Loss: 0.5358
Epoch 11/50, Training Loss: 0.5053, Validation Loss: 0.5471
Epoch 12/50, Training Loss: 0.5161, Validation Loss: 0.5598
Epoch 13/50, Training Loss: 0.5272, Validation Loss: 0.5723
Epoch 14/50, Training Loss: 0.5360, Validation Loss: 0.5824
Epoch 15/50, Training Loss: 0.5460, Validation Loss: 0.5934
Epoch 16/50, Training Loss: 0.5539, Validation Loss: 0.6025
Epoch 17/50, Training Loss: 0.5632, Validation Loss: 0.6127
Epoch 18/50, Training Loss: 0.5704, Validation Loss: 0.6210
Epoch 19/50, Training Loss: 0.5792, Validation Loss: 0.6305
Epoch 20/50, Training Loss: 0.5857, Validation Loss: 0.6382
Epoch 21/50, Training Loss: 0.5946, Validation Loss: 0.6480

Epoch 22/50, Training Loss: 0.6002, Validation Loss: 0.6545
Epoch 23/50, Training Loss: 0.6080, Validation Loss: 0.6630
Epoch 24/50, Training Loss: 0.6145, Validation Loss: 0.6703
Epoch 25/50, Training Loss: 0.6226, Validation Loss: 0.6796
Epoch 26/50, Training Loss: 0.6292, Validation Loss: 0.6872
Epoch 27/50, Training Loss: 0.6368, Validation Loss: 0.6953
Epoch 28/50, Training Loss: 0.6424, Validation Loss: 0.7016
Epoch 29/50, Training Loss: 0.6493, Validation Loss: 0.7093
Epoch 30/50, Training Loss: 0.6556, Validation Loss: 0.7166
Epoch 31/50, Training Loss: 0.6640, Validation Loss: 0.7249
Epoch 32/50, Training Loss: 0.6690, Validation Loss: 0.7310
Epoch 33/50, Training Loss: 0.6751, Validation Loss: 0.7377
Epoch 34/50, Training Loss: 0.6803, Validation Loss: 0.7435
Epoch 35/50, Training Loss: 0.6859, Validation Loss: 0.7491
Epoch 36/50, Training Loss: 0.6925, Validation Loss: 0.7559
Epoch 37/50, Training Loss: 0.6975, Validation Loss: 0.7614
Epoch 38/50, Training Loss: 0.7030, Validation Loss: 0.7672
Epoch 39/50, Training Loss: 0.7089, Validation Loss: 0.7734
Epoch 40/50, Training Loss: 0.7141, Validation Loss: 0.7787
Epoch 41/50, Training Loss: 0.7190, Validation Loss: 0.7840
Epoch 42/50, Training Loss: 0.7246, Validation Loss: 0.7897
Epoch 43/50, Training Loss: 0.7288, Validation Loss: 0.7943
Epoch 44/50, Training Loss: 0.7341, Validation Loss: 0.7996
Epoch 45/50, Training Loss: 0.7386, Validation Loss: 0.8044
Epoch 46/50, Training Loss: 0.7430, Validation Loss: 0.8088
Epoch 47/50, Training Loss: 0.7469, Validation Loss: 0.8131
Epoch 48/50, Training Loss: 0.7517, Validation Loss: 0.8179
Epoch 49/50, Training Loss: 0.7560, Validation Loss: 0.8225
Epoch 50/50, Training Loss: 0.7596, Validation Loss: 0.8263

Distribusi Bobot:



Distribusi Gradien Bobot:



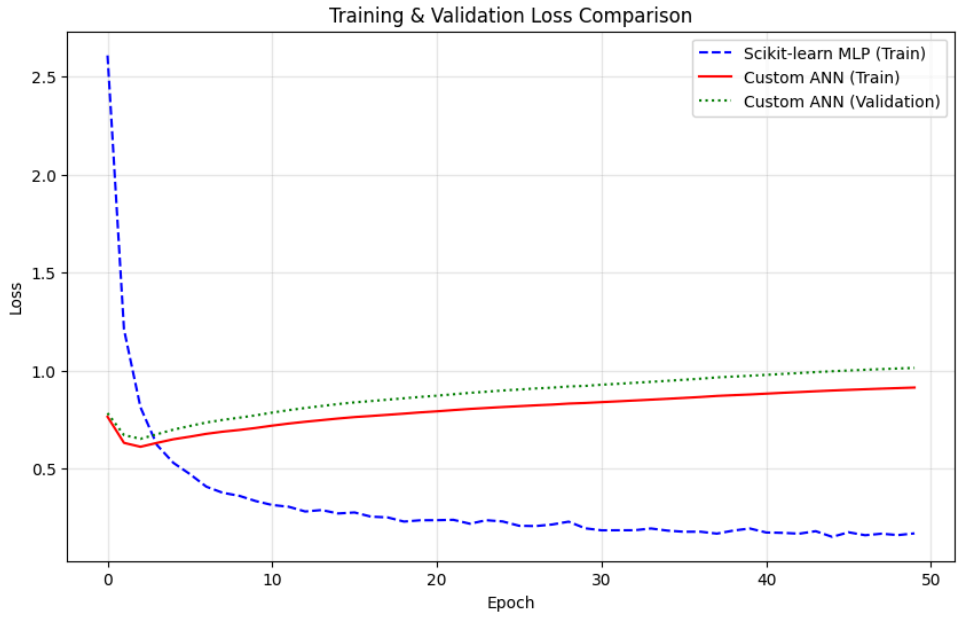
b. Regularisasi L1

Hasil prediksi

- Test Accuracy: 0.8191

```
Test Accuracy: 0.8191
```

Grafik loss

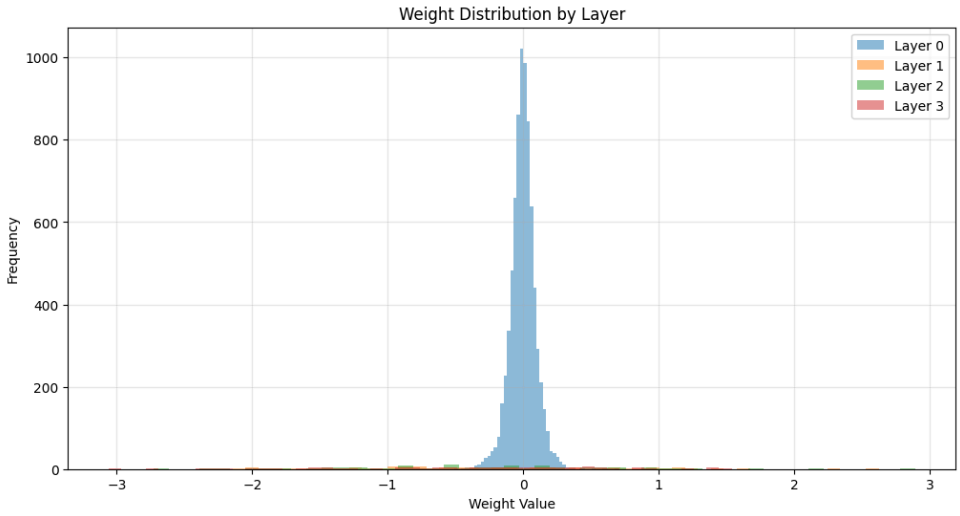


Iterasi Custom ANN:
Epoch 1/50, Training Loss: 0.7640, Validation Loss: 0.7823

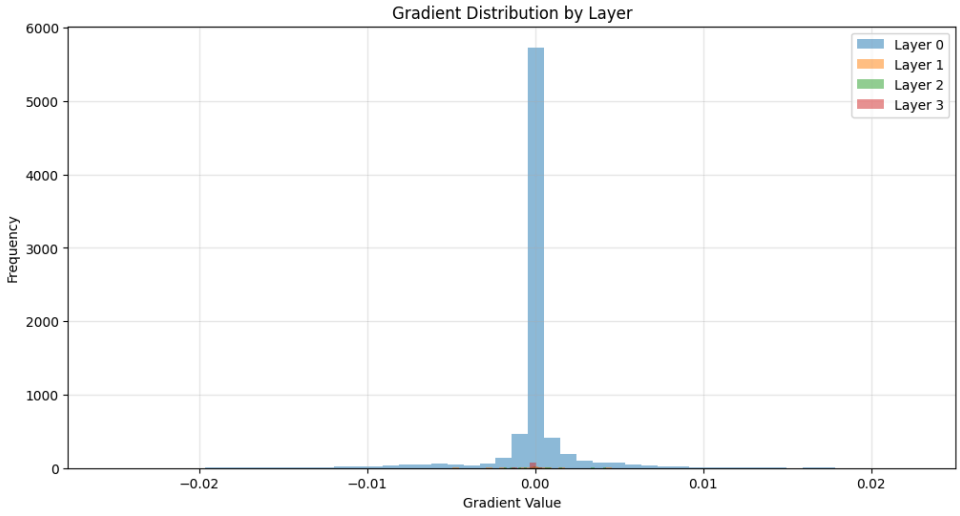
	Epoch 2/50, Training Loss: 0.6307, Validation Loss: 0.6705 Epoch 3/50, Training Loss: 0.6108, Validation Loss: 0.6513 Epoch 4/50, Training Loss: 0.6306, Validation Loss: 0.6747 Epoch 5/50, Training Loss: 0.6491, Validation Loss: 0.6983 Epoch 6/50, Training Loss: 0.6623, Validation Loss: 0.7169 Epoch 7/50, Training Loss: 0.6769, Validation Loss: 0.7347 Epoch 8/50, Training Loss: 0.6879, Validation Loss: 0.7481 Epoch 9/50, Training Loss: 0.6965, Validation Loss: 0.7591 Epoch 10/50, Training Loss: 0.7068, Validation Loss: 0.7712 Epoch 11/50, Training Loss: 0.7186, Validation Loss: 0.7852 Epoch 12/50, Training Loss: 0.7292, Validation Loss: 0.7977 Epoch 13/50, Training Loss: 0.7384, Validation Loss: 0.8086 Epoch 14/50, Training Loss: 0.7467, Validation Loss: 0.8188 Epoch 15/50, Training Loss: 0.7550, Validation Loss: 0.8288 Epoch 16/50, Training Loss: 0.7623, Validation Loss: 0.8374 Epoch 17/50, Training Loss: 0.7679, Validation Loss: 0.8440 Epoch 18/50, Training Loss: 0.7739, Validation Loss: 0.8511 Epoch 19/50, Training Loss: 0.7799, Validation Loss: 0.8582 Epoch 20/50, Training Loss: 0.7864, Validation Loss: 0.8654 Epoch 21/50, Training Loss: 0.7917, Validation Loss: 0.8715 Epoch 22/50, Training Loss: 0.7980, Validation Loss: 0.8786 Epoch 23/50, Training Loss: 0.8038, Validation Loss: 0.8858 Epoch 24/50, Training Loss: 0.8088, Validation Loss: 0.8917 Epoch 25/50, Training Loss: 0.8136, Validation Loss: 0.8980 Epoch 26/50, Training Loss: 0.8179, Validation Loss: 0.9030 Epoch 27/50, Training Loss: 0.8223, Validation Loss: 0.9085 Epoch 28/50, Training Loss: 0.8261, Validation Loss: 0.9123 Epoch 29/50, Training Loss: 0.8312, Validation Loss: 0.9186 Epoch 30/50, Training Loss: 0.8343, Validation Loss: 0.9213 Epoch 31/50, Training Loss: 0.8386, Validation Loss: 0.9275 Epoch 32/50, Training Loss: 0.8427, Validation Loss: 0.9321 Epoch 33/50, Training Loss: 0.8469, Validation Loss: 0.9372 Epoch 34/50, Training Loss: 0.8512, Validation Loss: 0.9422 Epoch 35/50, Training Loss: 0.8556, Validation Loss: 0.9474 Epoch 36/50, Training Loss: 0.8601, Validation Loss: 0.9527 Epoch 37/50, Training Loss: 0.8648, Validation Loss: 0.9582 Epoch 38/50, Training Loss: 0.8703, Validation Loss: 0.9648 Epoch 39/50, Training Loss: 0.8740, Validation Loss: 0.9690 Epoch 40/50, Training Loss: 0.8776, Validation Loss: 0.9730 Epoch 41/50, Training Loss: 0.8819, Validation Loss: 0.9777 Epoch 42/50, Training Loss: 0.8861, Validation Loss: 0.9825 Epoch 43/50, Training Loss: 0.8900, Validation Loss: 0.9871 Epoch 44/50, Training Loss: 0.8940, Validation Loss: 0.9918 Epoch 45/50, Training Loss: 0.8978, Validation Loss: 0.9961 Epoch 46/50, Training Loss: 0.9012, Validation Loss: 1.0000 Epoch 47/50, Training Loss: 0.9043, Validation Loss: 1.0036

Epoch 48/50, Training Loss: 0.9074, Validation Loss: 1.0070
Epoch 49/50, Training Loss: 0.9102, Validation Loss: 1.0102
Epoch 50/50, Training Loss: 0.9129, Validation Loss: 1.0132

Distribusi Bobot:



Distribusi Gradien Bobot:



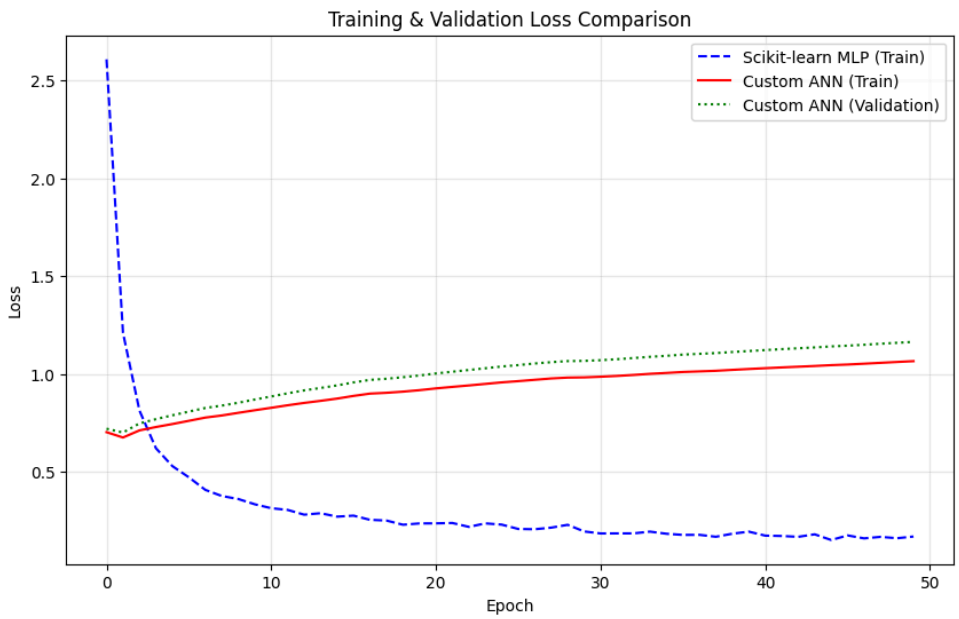
c. Regularisasi L2

Hasil prediksi

- Test Accuracy: 0.8254

```
Test Accuracy: 0.8254
```

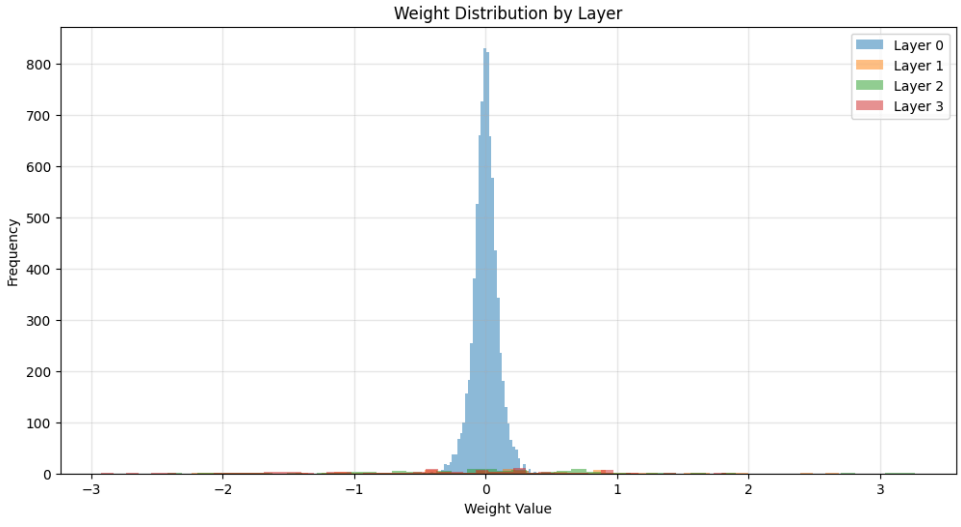
Grafik loss



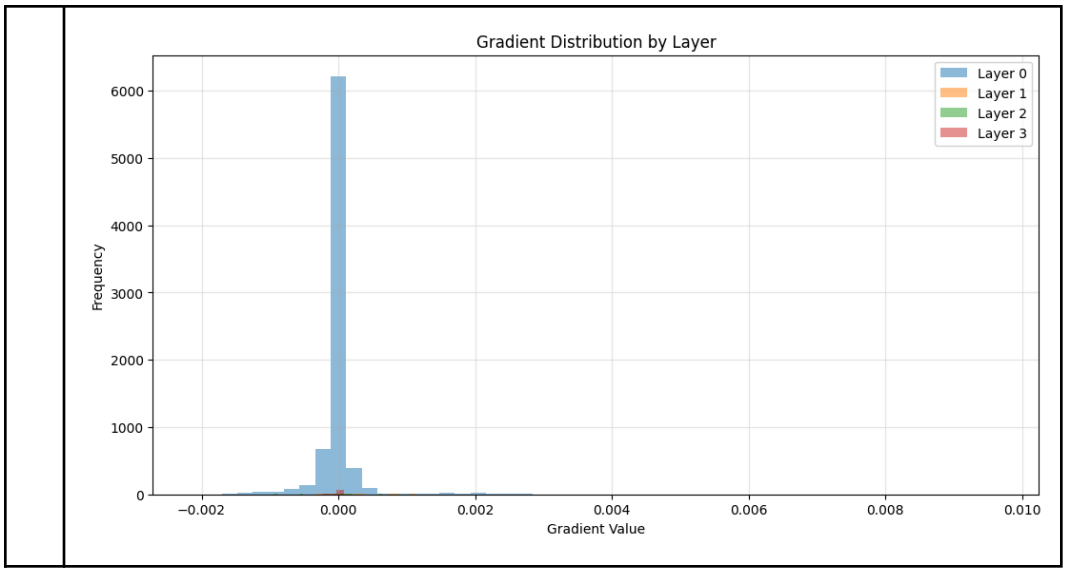
Iterasi Custom ANN:
Epoch 1/50, Training Loss: 0.7022, Validation Loss: 0.7198
Epoch 2/50, Training Loss: 0.6749, Validation Loss: 0.7004
Epoch 3/50, Training Loss: 0.7113, Validation Loss: 0.7467
Epoch 4/50, Training Loss: 0.7285, Validation Loss: 0.7688
Epoch 5/50, Training Loss: 0.7438, Validation Loss: 0.7885
Epoch 6/50, Training Loss: 0.7603, Validation Loss: 0.8073
Epoch 7/50, Training Loss: 0.7768, Validation Loss: 0.8258
Epoch 8/50, Training Loss: 0.7875, Validation Loss: 0.8381
Epoch 9/50, Training Loss: 0.8011, Validation Loss: 0.8525
Epoch 10/50, Training Loss: 0.8142, Validation Loss: 0.8690
Epoch 11/50, Training Loss: 0.8267, Validation Loss: 0.8846
Epoch 12/50, Training Loss: 0.8397, Validation Loss: 0.9007
Epoch 13/50, Training Loss: 0.8517, Validation Loss: 0.9158
Epoch 14/50, Training Loss: 0.8623, Validation Loss: 0.9282
Epoch 15/50, Training Loss: 0.8738, Validation Loss: 0.9411
Epoch 16/50, Training Loss: 0.8875, Validation Loss: 0.9566
Epoch 17/50, Training Loss: 0.8992, Validation Loss: 0.9694
Epoch 18/50, Training Loss: 0.9034, Validation Loss: 0.9752
Epoch 19/50, Training Loss: 0.9093, Validation Loss: 0.9830
Epoch 20/50, Training Loss: 0.9168, Validation Loss: 0.9918
Epoch 21/50, Training Loss: 0.9259, Validation Loss: 1.0022
Epoch 22/50, Training Loss: 0.9336, Validation Loss: 1.0112
Epoch 23/50, Training Loss: 0.9409, Validation Loss: 1.0200
Epoch 24/50, Training Loss: 0.9488, Validation Loss: 1.0288
Epoch 25/50, Training Loss: 0.9567, Validation Loss: 1.0379
Epoch 26/50, Training Loss: 0.9630, Validation Loss: 1.0450

Epoch 27/50, Training Loss: 0.9698, Validation Loss: 1.0532
Epoch 28/50, Training Loss: 0.9767, Validation Loss: 1.0599
Epoch 29/50, Training Loss: 0.9811, Validation Loss: 1.0654
Epoch 30/50, Training Loss: 0.9822, Validation Loss: 1.0665
Epoch 31/50, Training Loss: 0.9855, Validation Loss: 1.0700
Epoch 32/50, Training Loss: 0.9896, Validation Loss: 1.0751
Epoch 33/50, Training Loss: 0.9946, Validation Loss: 1.0810
Epoch 34/50, Training Loss: 1.0004, Validation Loss: 1.0873
Epoch 35/50, Training Loss: 1.0049, Validation Loss: 1.0923
Epoch 36/50, Training Loss: 1.0098, Validation Loss: 1.0986
Epoch 37/50, Training Loss: 1.0128, Validation Loss: 1.1024
Epoch 38/50, Training Loss: 1.0158, Validation Loss: 1.1064
Epoch 39/50, Training Loss: 1.0205, Validation Loss: 1.1119
Epoch 40/50, Training Loss: 1.0250, Validation Loss: 1.1168
Epoch 41/50, Training Loss: 1.0291, Validation Loss: 1.1218
Epoch 42/50, Training Loss: 1.0330, Validation Loss: 1.1265
Epoch 43/50, Training Loss: 1.0368, Validation Loss: 1.1309
Epoch 44/50, Training Loss: 1.0404, Validation Loss: 1.1353
Epoch 45/50, Training Loss: 1.0448, Validation Loss: 1.1405
Epoch 46/50, Training Loss: 1.0479, Validation Loss: 1.1440
Epoch 47/50, Training Loss: 1.0521, Validation Loss: 1.1489
Epoch 48/50, Training Loss: 1.0566, Validation Loss: 1.1541
Epoch 49/50, Training Loss: 1.0609, Validation Loss: 1.1589
Epoch 50/50, Training Loss: 1.0651, Validation Loss: 1.1636

Distribusi Bobot:



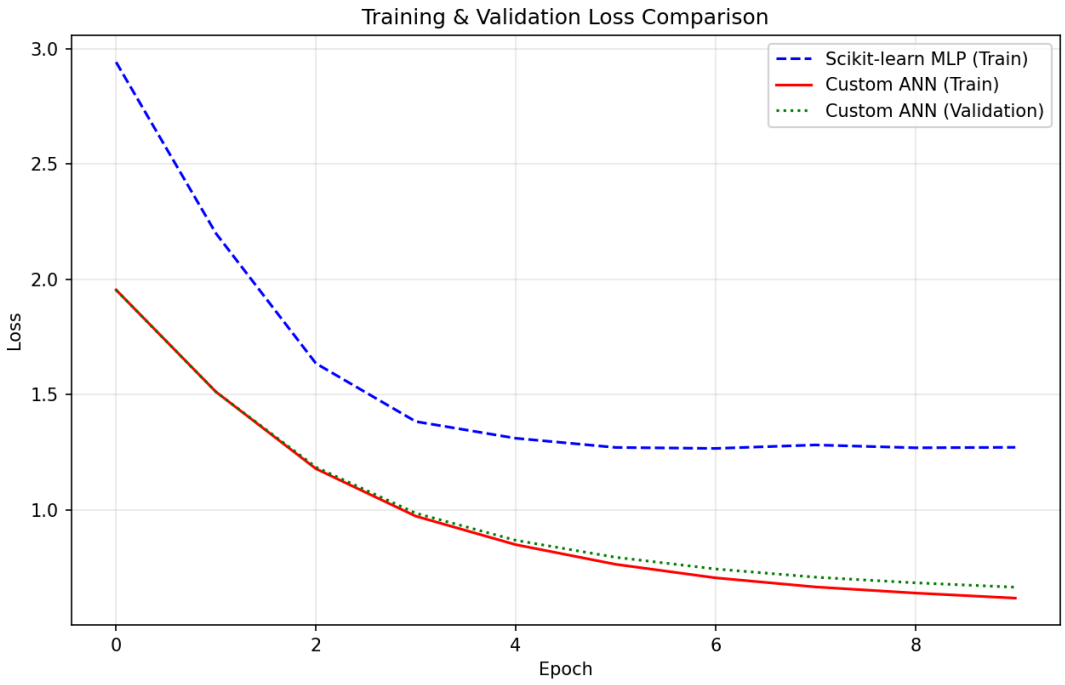
Distribusi Gradien Bobot:



Analisis:
Dari hasil pengujian di atas, dapat dilihat bahwa hasil akurasi test terbesar dicapai ketika model tidak menggunakan regularisasi. Akurasi terbesar kedua dicapai ketika model menggunakan regularisasi L2. Dari hasil tersebut dapat disimpulkan bahwa regularisasi memiliki pengaruh negatif terhadap performa model, meskipun hanya sedikit. Kesimpulan tersebut juga diperkuat dengan observasi bahwa nilai *loss* terkecil dicapai ketika model tidak menggunakan regularisasi. Namun, distribusi gradien dan bobot ketiga model pada semua layer relatif mirip, dengan ketiga model memiliki banyak bobot dan gradien dengan nilai 0.

6. Perbandingan dengan library sklearn MLP

Global Hyperparameter	
Depth	3
Width (All Layers)	10
Epoch	50
Seed/Random State	42
Learning Rate	0.001
Activation Function	Sigmoid
Batch Size	32
MLP	

<table>
<tr><td></td><td>Hasil Prediksi
Accuracy: 0.8829
<code>Accuracy: 0.8829</code></td></tr>
<tr><td colspan="2">Custom ANN</td></tr>
<tr><td rowspan="2"></td><td>Hyperparameter:
- Loss: Categorical Cross Entropy
- Initialization: Xavier</td></tr>
<tr><td>Hasil Prediksi
Test Accuracy: 0.8332
<code>Test Accuracy: 0.8332</code></td></tr>
<tr><td colspan="2">Grafik Loss</td></tr>
<tr><td colspan="2"> 
</td></tr>
</table>

Analisis:

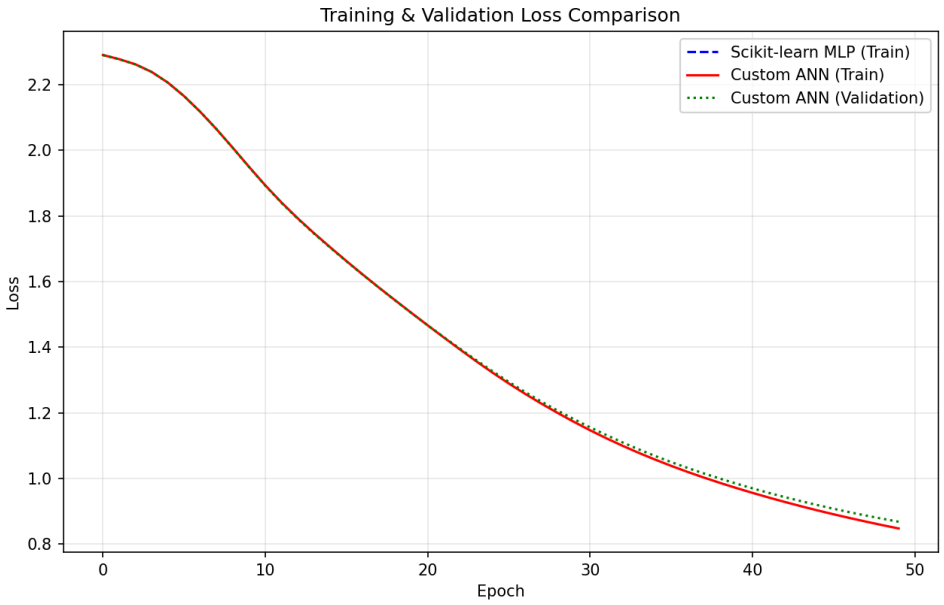
Dapat dilihat bahwa meskipun Custom ANN memiliki loss yang lebih kecil, MLP memiliki akurasi yang lebih tinggi. Hal ini menunjukkan bahwa Custom ANN memiliki kecenderungan untuk mengalami *overfitting*. Dapat dilihat juga bahwa grafik loss pada MLP lebih curam daripada Custom ANN. Hal ini menunjukkan bahwa proses pelatihan model pada MLP masih lebih efektif daripada Custom ANN.

7. Perbandingan metode normalisasi RMSNorm

Global Hyperparameter	
Depth	3
Width (All Layers)	10
Epoch	50
Seed/Random State	42
Learning Rate	0.001
Activation Function	Sigmoid
Loss	Categorical Cross Entropy
Batch Size	32
Regularisasi	None
Inisialisasi	Uniform: - Lower Bound: -1 - Upper Bound: 1 - Seed: 42

1. Tanpa normalisasi RMSNorm

	Hasil prediksi - Test Accuracy: 0.7669 `Test Accuracy: 0.7669`
	Grafik loss

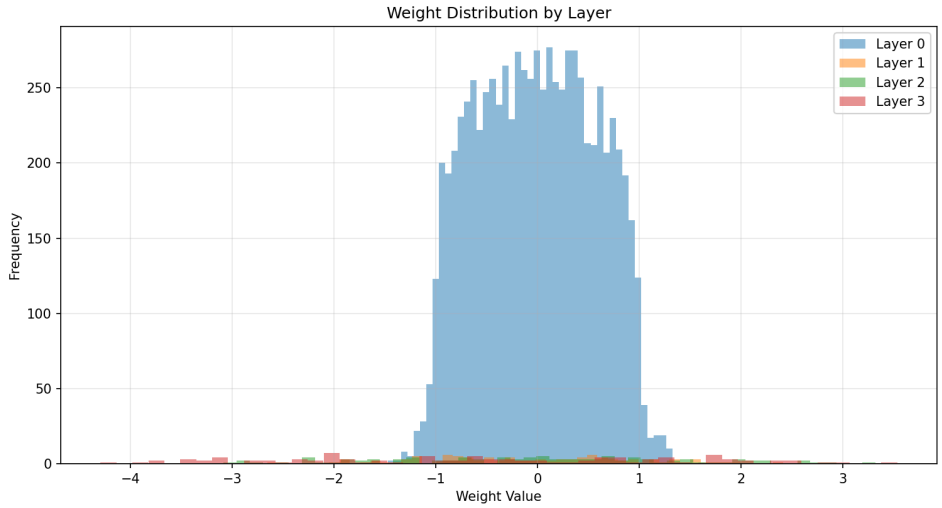


Iterasi Custom ANN:

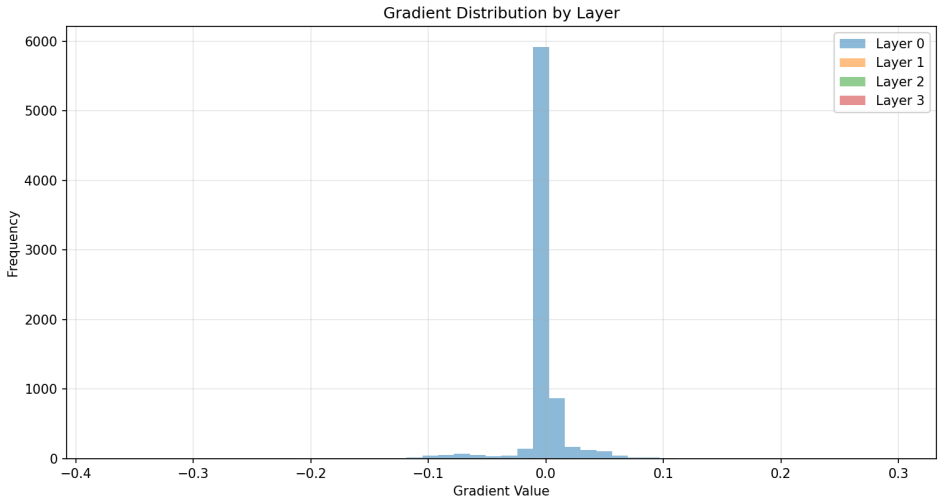
Epoch 1/50, Training Loss: 2.2904, Validation Loss: 2.2900
Epoch 2/50, Training Loss: 2.2778, Validation Loss: 2.2774
Epoch 3/50, Training Loss: 2.2620, Validation Loss: 2.2618
Epoch 4/50, Training Loss: 2.2388, Validation Loss: 2.2387
Epoch 5/50, Training Loss: 2.2063, Validation Loss: 2.2063
Epoch 6/50, Training Loss: 2.1649, Validation Loss: 2.1651
Epoch 7/50, Training Loss: 2.1169, Validation Loss: 2.1172
Epoch 8/50, Training Loss: 2.0640, Validation Loss: 2.0640
Epoch 9/50, Training Loss: 2.0075, Validation Loss: 2.0072
Epoch 10/50, Training Loss: 1.9495, Validation Loss: 1.9489
Epoch 11/50, Training Loss: 1.8932, Validation Loss: 1.8922
Epoch 12/50, Training Loss: 1.8408, Validation Loss: 1.8396
Epoch 13/50, Training Loss: 1.7926, Validation Loss: 1.7912
Epoch 14/50, Training Loss: 1.7474, Validation Loss: 1.7461
Epoch 15/50, Training Loss: 1.7044, Validation Loss: 1.7033
Epoch 16/50, Training Loss: 1.6624, Validation Loss: 1.6616
Epoch 17/50, Training Loss: 1.6214, Validation Loss: 1.6210
Epoch 18/50, Training Loss: 1.5816, Validation Loss: 1.5812
Epoch 19/50, Training Loss: 1.5428, Validation Loss: 1.5423
Epoch 20/50, Training Loss: 1.5045, Validation Loss: 1.5041
Epoch 21/50, Training Loss: 1.4666, Validation Loss: 1.4668
Epoch 22/50, Training Loss: 1.4294, Validation Loss: 1.4305
Epoch 23/50, Training Loss: 1.3927, Validation Loss: 1.3950
Epoch 24/50, Training Loss: 1.3568, Validation Loss: 1.3602
Epoch 25/50, Training Loss: 1.3223, Validation Loss: 1.3266
Epoch 26/50, Training Loss: 1.2893, Validation Loss: 1.2943
Epoch 27/50, Training Loss: 1.2579, Validation Loss: 1.2634

Epoch 28/50, Training Loss: 1.2281, Validation Loss: 1.2344
Epoch 29/50, Training Loss: 1.1998, Validation Loss: 1.2070
Epoch 30/50, Training Loss: 1.1727, Validation Loss: 1.1808
Epoch 31/50, Training Loss: 1.1469, Validation Loss: 1.1560
Epoch 32/50, Training Loss: 1.1226, Validation Loss: 1.1324
Epoch 33/50, Training Loss: 1.0997, Validation Loss: 1.1100
Epoch 34/50, Training Loss: 1.0781, Validation Loss: 1.0887
Epoch 35/50, Training Loss: 1.0577, Validation Loss: 1.0687
Epoch 36/50, Training Loss: 1.0384, Validation Loss: 1.0499
Epoch 37/50, Training Loss: 1.0201, Validation Loss: 1.0323
Epoch 38/50, Training Loss: 1.0029, Validation Loss: 1.0156
Epoch 39/50, Training Loss: 0.9866, Validation Loss: 0.9996
Epoch 40/50, Training Loss: 0.9711, Validation Loss: 0.9844
Epoch 41/50, Training Loss: 0.9563, Validation Loss: 0.9700
Epoch 42/50, Training Loss: 0.9420, Validation Loss: 0.9562
Epoch 43/50, Training Loss: 0.9284, Validation Loss: 0.9431
Epoch 44/50, Training Loss: 0.9154, Validation Loss: 0.9308
Epoch 45/50, Training Loss: 0.9028, Validation Loss: 0.9190
Epoch 46/50, Training Loss: 0.8909, Validation Loss: 0.9078
Epoch 47/50, Training Loss: 0.8794, Validation Loss: 0.8971
Epoch 48/50, Training Loss: 0.8685, Validation Loss: 0.8870
Epoch 49/50, Training Loss: 0.8579, Validation Loss: 0.8774
Epoch 50/50, Training Loss: 0.8478, Validation Loss: 0.8683

Distribusi Bobot:



Distribusi Gradien Bobot:



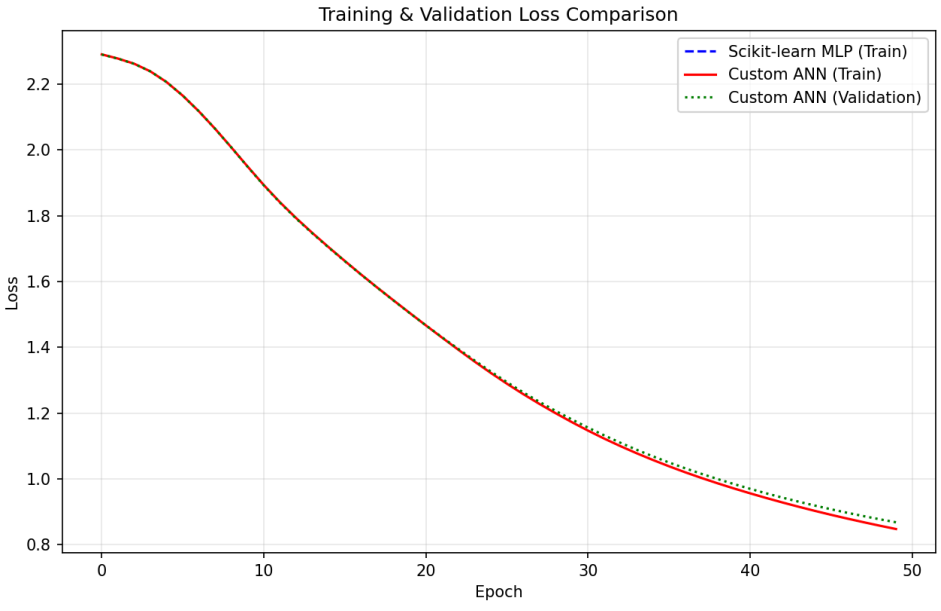
2. Dengan normalisasi RMSNorm

Hasil prediksi

- Test Accuracy: 0.7669

```
Test Accuracy: 0.7669
```

Grafik loss



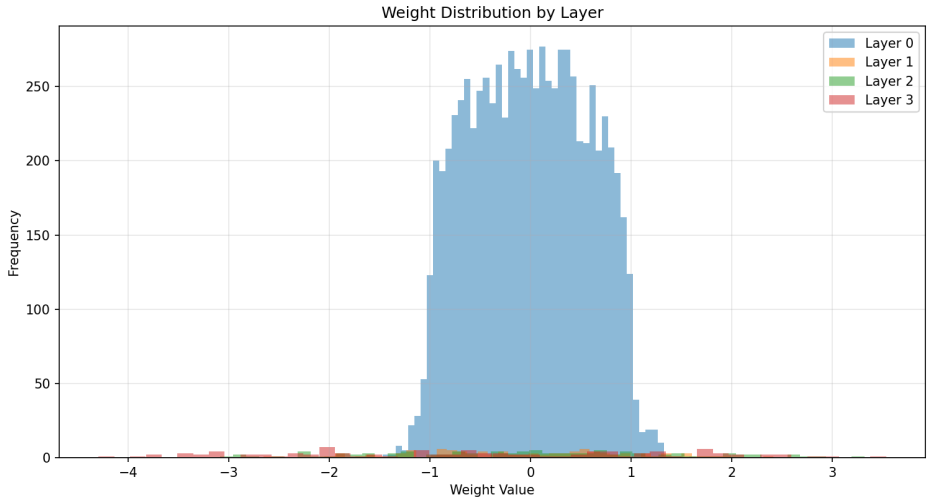
Iterasi Custom ANN:

Epoch 1/50, Training Loss: 2.2904, Validation Loss: 2.2900
Epoch 2/50, Training Loss: 2.2778, Validation Loss: 2.2774

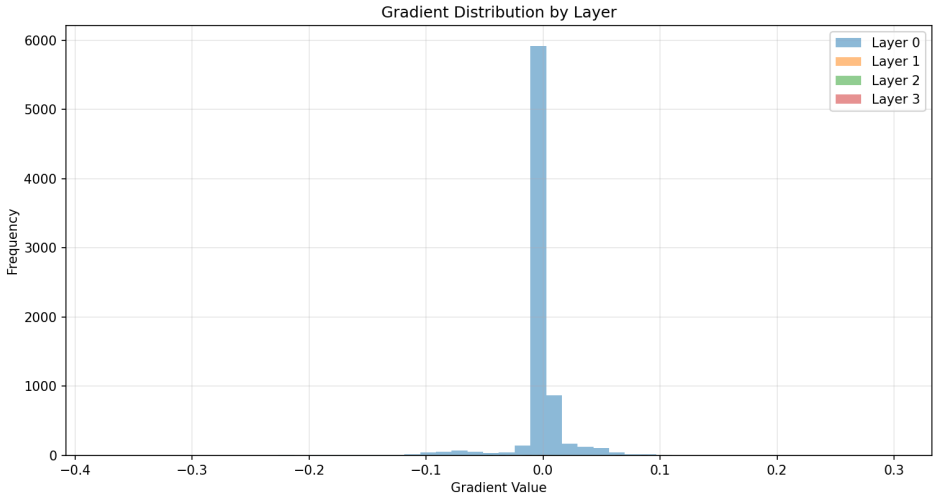
```
Epoch 3/50, Training Loss: 2.2620, Validation Loss: 2.2618
Epoch 4/50, Training Loss: 2.2388, Validation Loss: 2.2387
Epoch 5/50, Training Loss: 2.2063, Validation Loss: 2.2063
Epoch 6/50, Training Loss: 2.1649, Validation Loss: 2.1651
Epoch 7/50, Training Loss: 2.1169, Validation Loss: 2.1172
Epoch 8/50, Training Loss: 2.0640, Validation Loss: 2.0640
Epoch 9/50, Training Loss: 2.0075, Validation Loss: 2.0072
Epoch 10/50, Training Loss: 1.9495, Validation Loss: 1.9489
Epoch 11/50, Training Loss: 1.8932, Validation Loss: 1.8922
Epoch 12/50, Training Loss: 1.8408, Validation Loss: 1.8396
Epoch 13/50, Training Loss: 1.7926, Validation Loss: 1.7912
Epoch 14/50, Training Loss: 1.7474, Validation Loss: 1.7461
Epoch 15/50, Training Loss: 1.7044, Validation Loss: 1.7033
Epoch 16/50, Training Loss: 1.6624, Validation Loss: 1.6616
Epoch 17/50, Training Loss: 1.6214, Validation Loss: 1.6210
Epoch 18/50, Training Loss: 1.5816, Validation Loss: 1.5812
Epoch 19/50, Training Loss: 1.5428, Validation Loss: 1.5423
Epoch 20/50, Training Loss: 1.5045, Validation Loss: 1.5041
Epoch 21/50, Training Loss: 1.4666, Validation Loss: 1.4668
Epoch 22/50, Training Loss: 1.4294, Validation Loss: 1.4305
Epoch 23/50, Training Loss: 1.3927, Validation Loss: 1.3950
Epoch 24/50, Training Loss: 1.3568, Validation Loss: 1.3602
Epoch 25/50, Training Loss: 1.3223, Validation Loss: 1.3266
Epoch 26/50, Training Loss: 1.2893, Validation Loss: 1.2943
Epoch 27/50, Training Loss: 1.2579, Validation Loss: 1.2634
Epoch 28/50, Training Loss: 1.2281, Validation Loss: 1.2344
Epoch 29/50, Training Loss: 1.1998, Validation Loss: 1.2070
Epoch 30/50, Training Loss: 1.1727, Validation Loss: 1.1808
Epoch 31/50, Training Loss: 1.1469, Validation Loss: 1.1560
Epoch 32/50, Training Loss: 1.1226, Validation Loss: 1.1324
Epoch 33/50, Training Loss: 1.0997, Validation Loss: 1.1100
Epoch 34/50, Training Loss: 1.0781, Validation Loss: 1.0887
Epoch 35/50, Training Loss: 1.0577, Validation Loss: 1.0687
Epoch 36/50, Training Loss: 1.0384, Validation Loss: 1.0499
Epoch 37/50, Training Loss: 1.0201, Validation Loss: 1.0323
Epoch 38/50, Training Loss: 1.0029, Validation Loss: 1.0156
Epoch 39/50, Training Loss: 0.9866, Validation Loss: 0.9996
Epoch 40/50, Training Loss: 0.9711, Validation Loss: 0.9844
Epoch 41/50, Training Loss: 0.9563, Validation Loss: 0.9700
Epoch 42/50, Training Loss: 0.9420, Validation Loss: 0.9562
Epoch 43/50, Training Loss: 0.9284, Validation Loss: 0.9431
Epoch 44/50, Training Loss: 0.9154, Validation Loss: 0.9308
Epoch 45/50, Training Loss: 0.9028, Validation Loss: 0.9190
Epoch 46/50, Training Loss: 0.8909, Validation Loss: 0.9078
Epoch 47/50, Training Loss: 0.8794, Validation Loss: 0.8971
Epoch 48/50, Training Loss: 0.8685, Validation Loss: 0.8870
```

Epoch 49/50, Training Loss: 0.8579, Validation Loss: 0.8774
Epoch 50/50, Training Loss: 0.8478, Validation Loss: 0.8683

Distribusi Bobot:



Distribusi Gradien Bobot:



Analisis:

Dapat dilihat bahwa custom ANN tanpa normalisasi RMSNorm memiliki hasil akurasi terhadap dataset *test* sebesar 76,69%, sama dengan hasil loss custom ANN dengan normalisasi RMSNorm. Grafik yang dihasilkan dari kedua model juga memiliki bentuk yang mirip. Hal ini menunjukkan bahwa pengaruh normalisasi RMSNorm terhadap akurasi kinerja custom ANN sangat kecil. Bentuk distribusi bobot dan gradien kedua model di setiap layer juga sangat mirip. Dapat disimpulkan bahwa normalisasi RMSNorm memiliki pengaruh kecil terhadap kinerja custom ANN.

BAB 3
KESIMPULAN DAN SARAN

3.1. Kesimpulan

Laporan ini memberikan penjelasan komprehensif mengenai implementasi dan pengujian *Feedforward Neural Network* (FFNN) dari awal (*scratch*). Model yang diimplementasikan diuji menggunakan dataset MNIST, dan dari hasil eksperimen dapat disimpulkan hal berikut:

1. Arsitektur FFNN yang lebih dalam (*layer* lebih banyak) menurunkan performa model, yang ditunjukkan dengan nilai akurasi yang lebih rendah dan nilai *loss* yang lebih tinggi. Hal ini bisa terjadi karena terjadinya *overfitting* ketika *layer* model bertambah.
2. Arsitektur FFNN yang lebih lebar (jumlah neuron lebih banyak) menaikkan performa model, yang ditunjukkan dengan nilai akurasi yang lebih tinggi dan nilai *loss* yang lebih rendah.
3. Fungsi aktivasi sigmoid secara umum memberikan nilai akurasi lebih tinggi dibandingkan fungsi aktivasi lainnya. Hal ini bisa disebabkan karena fungsi aktivasi sigmoid memiliki rentang nilai antara 0 dan 1, yang mencegah terjadinya *vanishing/exploding gradient* pada model.
4. Inisialisasi bobot dengan metode Xavier secara umum memberikan performa model yang lebih tinggi dibandingkan metode inisialisasi lainnya (*zero, uniform, normal*, He).
5. Visualisasi distribusi bobot dan gradien memberikan wawasan tambahan terhadap proses kinerja model.

3.2. Saran

1. Untuk meningkatkan kinerja model dan meningkatkan akurasi, dapat digunakan metode *dropout*. Metode tersebut telah terbukti mengurangi terjadinya *overfitting* dan meningkatkan generalisasi model.
2. Validasi yang dilakukan dapat menggunakan metode lain seperti *cross-validation*. Validasi dengan *cross-validation* cenderung lebih stabil terhadap variasi data dibandingkan validasi dengan pemisahan *dataset* biasa.
3. Hasil prediksi model dan label sebenarnya dapat di-*plot* menjadi sebuah *heatmap* untuk menyoroti *error* pada kinerja model.

LAMPIRAN

PEMBAGIAN TUGAS

Nim	Nama	Tugas
13522002	**Ariel Herfrison**	**Melakukan pengujian, Debugging algoritma**
13522024	**Kristo Anugrah**	**Menyusun penjelasan implementasi, Debugging algoritma**
13522038	**Francesco Michael Kusuma**	**Melakukan pengujian, Membuat algoritma awal pelatihan model**

REFERENSI

Mitchell, T. M. (1997). Machine learning (Vol. 1). McGraw-hill New York.

Repository Github:

https://github.com/FrancescoMichael/Tubes1_ML