

# Laporan Eksplorasi Hyperparameter CNN & NN

Tugas Mata Kuliah Pembelajaran Mesin Lanjut

**Oleh:**

**Reza Budiawan – 33221040**



INSTITUT TEKNOLOGI BANDUNG

**Maret 2022**

# Daftar Isi

1.1.	Kasus 1: Klasifikasi—Convolutional Neural Network (CNN) .....	1
1.1.1.	Observasi.....	1
1.1.2.	Parameter yang Digunakan.....	6
1.1.3.	Model CNN.....	7
1.1.4.	Inferensi Data & Hasil .....	8
1.2.	Kasus 2: Regresi—Neural Network (NN).....	10
1.2.1.	Observasi.....	10
1.2.2.	Parameter yang Digunakan.....	12

# Daftar Tabel

Tabel 1: Hasil Akurasi terhadap Nilai Epoch .....	1
Tabel 2: Skenario Uji .....	2
Tabel 3: Nilai Akurasi terhadap Nilai Epoch per Skenario.....	2
Tabel 4: Skenario Pengujian Jumlah Layer Konvolusi .....	3
Tabel 5: Hasil Observasi Jumlah Layer .....	3
Tabel 6: Skenario Penentuan Ukuran Layer Konvolusi .....	3
Tabel 7: Hasil Eksperimen Penentuan Ukuran Layer Konvolusi.....	4
Tabel 8: Skenario Penentuan Jumlah Layer .....	4
Tabel 9: Eksperimen Jumlah Layer Konvolusi .....	5
Tabel 10: Hasil Akurasi terhadap Jumlah Unit FCN.....	5
Tabel 11: Hasil Eksperimen Akurasi terhadap Learning Rate .....	6
Tabel 12: Nilai MAE terhadap Nilai Epoch .....	10
Tabel 13: Eksperimen Jumlah Layer dan Jumlah Unit terhadap MAE .....	11
Tabel 14: Hasil Eksperimen Pengubahan Fungsi Aktivasi .....	11

# Daftar Gambar

Gambar 1: Plot Model CNN.....	7
Gambar 2: Ilustrasi Model .....	8
Gambar 3: Data Gagal Dikenali Sistem .....	8
Gambar 4: Data Berhasil Dikenali Sistem .....	8

# Eksplorasi Hyperparameter

Convolutional Neural Network (CNN) & Neural Network (NN)

## 1.1. Kasus 1: Klasifikasi—Convolutional Neural Network (CNN)

Kasus yang digunakan merupakan persoalan klasifikasi. Dataset yang digunakan adalah CIFAR10 yang terdiri dari 50.000 data latih dan 10.000 data uji. Data gambar diklasifikasikan pada 10 kategori: "airplane", "automobile", "bird", "cat", "deer", "dog", "frog", "horse", "ship", "truck".

### 1.1.1. Observasi

#### Penentuan nilai epoch

Observasi yang dilakukan pertama kali dengan menggunakan 2 layer konvolusi, dengan masing-masing layer berukuran 3x3 sebanyak 32. Layer maxpool digunakan dengan ukuran 2x2 dengan stride 2. Selain itu terdapat 1 dense layer dengan jumlah unit sebanyak 32.

Hyperparameter lain yang ditetapkan adalah optimizer Adam dengan learning rate yang sudah dimodifikasi, yaitu 0.01 (nilai default=0.001). Pada observasi ini, dilakukan perubahan epoch sebanyak 10 kali, dimulai dari nilai 10 hingga 100, dengan kenaikan 10. Hasil diperlihatkan pada tabel 1.

*Tabel 1: Hasil Akurasi terhadap Nilai Epoch*

Epoch	10	20	30	40	50	60	70	80	90	100
Nilai akurasi	47%	48%	49%	49%	48%	48%	48%	48%	46%	47%

Nilai yang dihasilkan, terdapat kenaikan pada epoch bernilai 30 & 40. Setelah epoch 40, terdapat penurunan akurasi.

Untuk melihat padakah trend ini berlaku juga untuk kasus lain, maka dilakukan juga eksperimen untuk melihat trend nilai epoch terhadap akurasi pada skenario yang berbeda. Skenario dituliskan pada tabel 2. Setiap skenario dilatih menggunakan optimizer Adam dengan nilai learning rate yang sudah dimodifikasi, yaitu 0.01 (nilai default=0.001). Masing-masing skenario dijalankan dalam 10 percobaan epoch, dimulai dari nilai 10 hingga 100 dengan kenaikan 10.

Tabel 2: Skenario Uji

Skenario A1	Skenario A2	Skenario A3
1 layer konvolusi: <ul style="list-style-type: none"> <li>Ukuran 3x3</li> <li>Jumlah: 32</li> <li>Stride: 1</li> </ul> 1 maxpool layer <ul style="list-style-type: none"> <li>Ukuran 2x2</li> <li>Stride: 2</li> </ul> 1 dense layer – 32 unit	2 layer konvolusi: <ul style="list-style-type: none"> <li>Ukuran layer 1: 3x3</li> <li>Jumlah layer 1: 32</li> <li>Ukuran layer 2: 3x3</li> <li>Jumlah layer 1: 32</li> <li>Stride: 1</li> </ul> 2 maxpool layer <ul style="list-style-type: none"> <li>Ukuran 2x2</li> <li>Stride: 2</li> </ul> 1 dense layer – 32 unit	2 layer konvolusi: <ul style="list-style-type: none"> <li>Ukuran layer 1: 3x3</li> <li>Jumlah layer 1: 32</li> <li>Ukuran layer 2: 3x3</li> <li>Jumlah layer 1: 64</li> <li>Stride: 1</li> </ul> 2 maxpool layer <ul style="list-style-type: none"> <li>Ukuran 2x2</li> <li>Stride: 2</li> </ul> 1 dense layer – 32 unit

Hasil percobaan ini didapat sebagai berikut, dengan nilai berupa persentase akurasi dari tiap skenario dapat dilihat pada tabel 3.

Tabel 3: Nilai Akurasi terhadap Nilai Epoch per Skenario

Epoch	10	20	30	40	50	60	70	80	90	100
Skenario A1	47%	48%	49%	49%	48%	48%	48%	48%	46%	47%
Skenario A2	41%	10%	10%	53%	48%	49%	10%	56%	53%	51%
Skenario A3	10%	10%	52%	53%	53%	53%	55%	53%	53%	55%

Beberapa hasil akurasi menunjukkan anomali berupa penurunan nilai (pada pengujian skenario A2). Akan tetapi trend yang didapat adalah naiknya nilai pada epoch bernilai 30 dan 40. Sedangkan pada epoch ke 40, 70 & 80 terdapat kenaikan nilai dan penurunan kembali setelahnya. Untuk itu, pada parameter epoch yang digunakan selanjutnya berkisar pada nilai 10-40.

### Penentuan jumlah layer konvolusi

Terdapat tiga skenario yang digunakan pada penentuan jumlah layer konvolusi. Hal ini dikarenakan ketika terjadi penambahan layer konvolusi keempat, terdapat error pada saat running. Skenario ini sebagian telah dilakukan pada eksperimen dalam menentukan jumlah epoch yang digunakan. Skenario pertama dan kedua dalam penentuan jumlah layer, memiliki konfigurasi yang sama seperti skenari A1 & A2 yang dituliskan pada Tabel 1. Sedangkan skenario tambahan berupa menggunakan 3 layer konvolusi dengan masing-masing ukuran layer 3x3 berjumlah 32. Parameter untuk maxpool dan dense layer disamakan.

Skenario ini dituliskan pada Tabel 4.

Tabel 4: Skenario Pengujian Jumlah Layer Konvolusi

Skenario B1	Skenario B2	Skenario B3
1 layer konvolusi: <ul style="list-style-type: none"> <li>Ukuran 3x3</li> <li>Jumlah: 32</li> <li>Stride: 1</li> </ul> 1 maxpool layer <ul style="list-style-type: none"> <li>Ukuran 2x2</li> <li>Stride: 2</li> </ul> 1 dense layer – 32 unit	2 layer konvolusi: <ul style="list-style-type: none"> <li>Ukuran layer 1: 3x3</li> <li>Jumlah layer 1: 32</li> <li>Ukuran layer 2: 3x3</li> <li>Jumlah layer 1: 32</li> <li>Stride: 1</li> </ul> 2 maxpool layer <ul style="list-style-type: none"> <li>Ukuran 2x2</li> <li>Stride: 2</li> </ul> 1 dense layer – 32 unit	2 layer konvolusi: <ul style="list-style-type: none"> <li>Ukuran layer 1: 3x3</li> <li>Jumlah layer 1: 32</li> <li>Ukuran layer 2: 3x3</li> <li>Jumlah layer 1: 32</li> <li>Ukuran layer 2: 3x3</li> <li>Jumlah layer 1: 32</li> <li>Stride: 1</li> </ul> 3 maxpool layer <ul style="list-style-type: none"> <li>Ukuran 2x2</li> <li>Stride: 2</li> </ul> 1 dense layer – 32 unit

Percobaan dilakukan pada epoch 10-40 saja. Hasil dari eksperimen ini dituliskan pada Tabel 5.

Tabel 5: Hasil Observasi Jumlah Layer

Epoch	10	20	30	40
Skenario B1	47%	48%	49%	49%
Skenario B2	41%	10%	10%	53%
Skenario B3	10%	48%	10%	48%

Hasil tertinggi yang didapat terdapat pada skenario B, yaitu sebesar 53%. Skenario B2 menggunakan 2 layer konvolusi. Sehingga, pada eksperimen berikutnya akan menggunakan 2 layer konvolusi saja.

#### Penentuan ukuran layer konvolusi

Dalam penentuan ukuran layer konvolusi, nilai yang biasa digunakan adalah 3,5,7, dan 9. Pada eksperimen kali ini hanya digunakan nilai 3,5, dan 7 saja. Ukuran layer konvolusi ditentukan dengan melakukan eksperimen menggunakan skenario yang dituliskan pada Tabel 6.

Tabel 6: Skenario Penentuan Ukuran Layer Konvolusi

Skenario C1	Skenario C2	Skenario C3
Layer 1: 32 @3x3 Layer 2: 32 @3x3	Layer 1: 32@5x5 Layer 2: 32@5x5	Layer 1: 32@7x7 Layer 2: 32@7x7
Skenario C4	Skenario C5	Skenario C6
Layer 1: 32 @7x7 Layer 2: 32 @5x5	Layer 1: 32@7x7 Layer 2: 32@3x3	Layer 1: 32@5x5 Layer 2: 32@3x3

Setiap skenario di atas, memiliki masing-masing maxpool yang seragam, yaitu berukuran 2x2 dengan stride 2. Learning rate yang digunakan 0.01 dengan optimizer Adam. Selain itu, dilakukan masing-masing 4x percobaan tergantung nilai epoch. Hasil eksperimen penentuan besar ukuran layer konvolusi & kombinasinya diperlihatkan pada Tabel 7.

*Tabel 7: Hasil Eksperimen Penentuan Ukuran Layer Konvolusi*

Epoch	10	20	30	40
<b>Skenario C1</b>	41%	10%	10%	<b>53%</b>
<b>Skenario C2</b>	40%	10%	44%	10%
<b>Skenario C3</b>	10%	10%	10%	10%
<b>Skenario C4</b>	10%	10%	10%	36%
<b>Skenario C5</b>	10%	32%	10%	41%
<b>Skenario C6</b>	10%	44%	47%	39%

Berdasarkan hasil eksperimen, terlihat konfigurasi C1 (ukuran layer konvolusi 3x3) mendapat nilai akurasi yang tertinggi. Sehingga, untuk berikutnya digunakan ukuran layer konvolusi 3x3.

#### **Penentuan jumlah layer konvolusi**

Eksperimen dalam menentukan jumlah layer yang optimal menggunakan konfigurasi berupa 2 layer konvolusi 3x3 (stride 1) dan maxpool 2x2 (stride 2). Epoch bernilai 40 pada saat melakukan eksperimen terkait penentuan jumlah layer, dikarenakan epoch ke-40 memberi nilai maksimal pada setiap skenario pada percobaan sebelumnya. Dari beberapa eksperimen yang pernah dilakukan terkait CNN, jumlah feature map semakin meningkat dari layer 1 menuju layer 2. Jadi pada eksperimen kali ini, ditentukan bahwa jumlah layer 1 lebih sedikit dibandingkan layer kedua.

Terdapat 3 nilai kombinasi yang digunakan: 32, 48, dan 64. Skenario yang digunakan dapat dilihat pada Tabel 8.

*Tabel 8: Skenario Penentuan Jumlah Layer*

<b>Skenario D1</b>	<b>Skenario D2</b>	<b>Skenario D3</b>
Layer 1: 32 @3x3 Layer 2: 32 @3x3	Layer 1: 32 @3x3 Layer 2: 48 @3x3	Layer 1: 32 @3x3 Layer 2: 64 @3x3
<b>Skenario D4</b>	<b>Skenario D5</b>	<b>Skenario D6</b>
Layer 1: 48 @3x3 Layer 2: 48 @3x3	Layer 1: 48 @3x3 Layer 2: 64 @3x3	Layer 1: 64 @3x3 Layer 2: 64 @3x3



Eksperimen yang dilakukan masih menggunakan Adam optimizer dengan learning rate 0.01. Hasil dari eksperimen ditampilkan pada Tabel 9.

*Tabel 9: Eksperimen Jumlah Layer Konvolusi*

Skenario	D1	D2	D3	D4	D5	D6
Akurasi	52%	56%	52%	53%	52%	10%

Hasil pengujian skenario D1 mengalami penurunan saat diuji kembali. Untuk hasil tertinggi, didapat dari skenario D2, yaitu nilai akurasi sebesar 56%. Sehingga, pada eksperimen berikutnya akan digunakan konfigurasi jumlah layer pertama sebanyak 32 layer, dan jumlah layer kedua sebanyak 48 layer.

#### **Penentuan jumlah unit FCN**

Fully connected network (FCN) merupakan salah satu layer yang digunakan pada arsitektur CNN dengan variasi jumlah yang beragam. Akan tetapi, pada eksperimen kali ini hanya digunakan 1 layer FCN. Sehingga, pencarian nilai maksimal terhadap nilai akurasi dikhususkan pada pencarian jumlah unit hidden layer saja. Variasi nilai hidden layer yang digunakan yaitu 32, 64, 128, 512, dan 1024.

Hyperparameter lain menggunakan skenario yang didapat dari eksperimen sebelumnya:

- 2 layer konvolusi: 32 @3x3, dan 48 @3x3; stride = 1
- 2 maxpool (2x2, stride = 2)
- Adam optimizer, learning rate 0.01
- Jumlah epoch: 40

Hasil eksperimen dari penentuan jumlah unit FCN diperlihatkan pada Tabel 10.

*Tabel 10: Hasil Akurasi terhadap Jumlah Unit FCN*

Nilai unit	32	64	128	512	1024
Akurasi	55%	55%	10%	46%	44%

Hasil pengujian kembali terhadap nilai unit 32 mengalami penurunan dibandingkan eksperimen sebelumnya. Akan tetapi hasil yang diberikan tidak terlalu jauh berbeda. Nilai akurasi tertinggi didapat pada eksperimen yang menggunakan 64 unit pada fully connected network. Untuk unit berjumlah 32 dan 64, memiliki persentasenya yang sama (55%). Walaupun demikian, unit berjumlah 64 sedikit lebih tinggi dibandingkan 32.

### Penentuan nilai learning rate

Learning rate pada Keras sudah menjadi bagian dari Optimizer yang digunakan. Akan tetapi, untuk melihat pengaruhnya, nilai learning rate diubah mulai dari 0.01 hingga 0.001 dengan perbedaan nilai sebesar 0.001 antara satu nilai dengan nilai sebelumnya. Hasil yang didapat dituliskan pada Tabel 11.

*Tabel 11: Hasil Eksperimen Akurasi terhadap Learning Rate*

Learning rate	0.001	0.002	0.003	0.004	0.005	0.006	0.007	0.008	0.009	0.010
Akurasi	71%	68%	65%	63%	60%	57%	49%	10%	10%	45%

Berdasarkan eksperimen yang dilakukan, didapat 0.001 mendapat nilai akurasi yang maksimal. Nilai learning rate 0.001 juga merupakan nilai default dari learning rate yang terdapat pada optimizer Adam dari library Keras.

### Penentuan Optimizer

Terdapat beberapa optimizer yang digunakan pada eksperimen ini. Hasil dari akurasi berdasarkan optimizer-nya, diperlihatkan pada Tabel 12.

Optimizer	SGD	RMSProp	Adam	Adadelta	Adagrad	Adamax	Nadam	Ftrl
Akurasi	71%	38%	69%	29%	48%	73%	71%	10%

#### 1.1.2. Parameter yang Digunakan

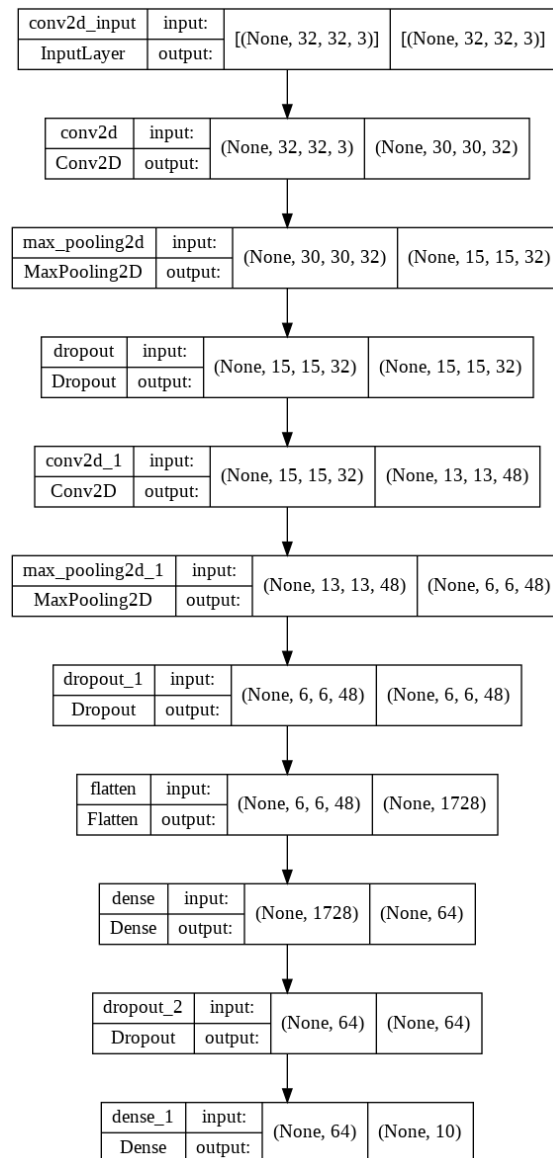
Berdasarkan hasil observasi, berikut parameter yang digunakan pada klasifikasi image menggunakan dataset CIFAR10.

Layer Konvolusi & Softmax	Epoch	Unit FCN	Optimizer
2 layer konvolusi dengan stride bernilai 1: 1) 32 @3x3 2) 48 @3x3  Tiap layer konvolusi berpasangan dengan maxpool layer: 2x2 dengan stride 2.	40	64	Adamax (learning rate = 0.001)

Akurasi akhir dari model yang digunakan adalah 73.71%. Hal ini sudah ditambahkan dengan dropout layer pada antar layer konvolusi & antar dense-softmax. Walaupun akurasi training telah mencapai lebih dari 80%, akurasi testing hanya mencapai nilai pembulatan 73%. Diperlukan eksplorasi lebih lanjut tentang bagaimana meningkatkan akurasi test, seperti augmentasi data dan trial-error kombinasi lain dari hyperparameter.

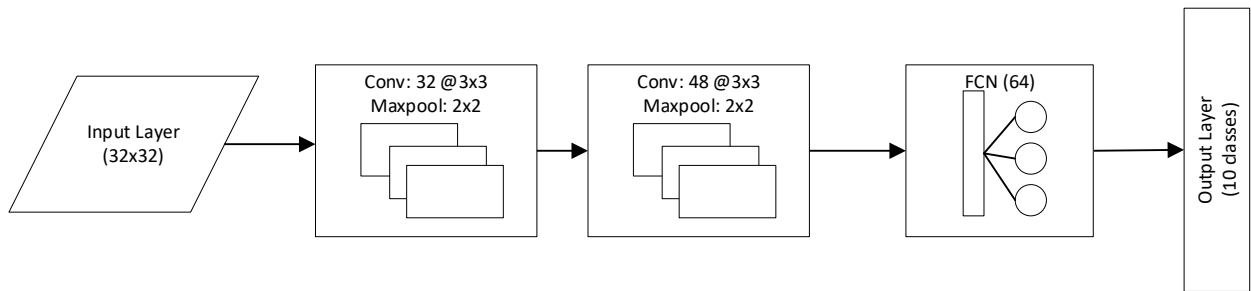
### 1.1.3. Model CNN

Model CNN yang dihasilkan diperlihatkan pada Gambar 1.



Gambar 1: Plot Model CNN

Ilustrasi dari cara kerja model diperlihatkan pada Gambar 2.

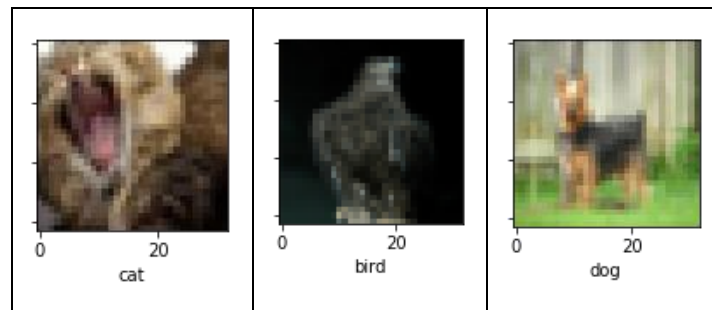


Gambar 2: Ilustrasi Model

Adamax

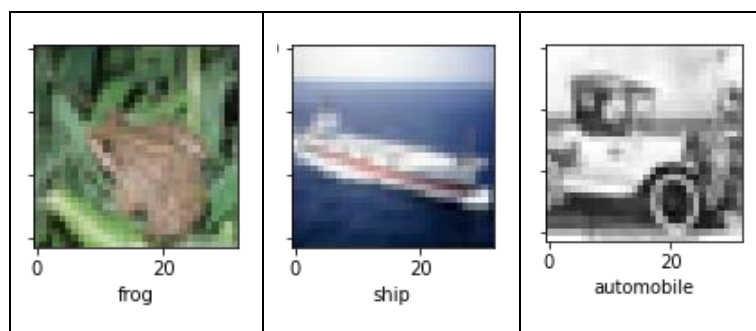
#### 1.1.4. Inferensi Data & Hasil

Hasil dari model yang dibuat memiliki akurasi testing senilai 72%. Contoh dari pengenalan yang gagal dikenali dengan benar oleh sistem adalah data berikut:



Gambar 3: Data Gagal Dikenali Sistem

Sedangkan contoh data yang berhasil dikenali benar oleh sistem diperlihatkan sebagai berikut:



Gambar 4: Data Berhasil Dikenali Sistem

Statistik dari data yang gagal diidentifikasi oleh CNN adalah sebagai berikut:

Label	Jum Gagal Prediksi	Kategori
0	272	airplane
1	142	automobile
2	425	bird
3	506	cat
4	282	dee
5	316	dog
6	125	frog
7	231	horse
8	162	ship
9	168	truck
<b>Total</b>		<b>2629</b>

Kategori cat, bird, dan dog merupakan kategori yang paling banyak gagal diprediksi oleh CNN. Sedangkan frog, ship, dan automobile merupakan kategori yang paling sedikit gagal diprediksi sistem.

## 1.2. Kasus 2: Regresi—Neural Network (NN)

Persoalan regresi merupakan hal yang diangkat pada kasus kedua. Dataset yang digunakan adalah Boston Housing Price. Persoalan regresi ini dapat dipecahkan menggunakan neural network.

### 1.2.1. Observasi

Observasi awal yang dilakukan adalah menentukan jumlah epoch yang digunakan dalam melakukan training.

#### **Menentukan nilai epoch**

Observasi dilakukan dengan menggunakan 2 hidden layer dengan masing-masing unit berjumlah 64. Optimizer yang digunakan adalah RMSprop, dan loss function adalah mean square error (MSE). Metric evaluation yang digunakan merupakan mean absolute error (MAE). Semakin kecil nilai MAE, maka akan semakin “bagus” nilai prediksi dari regresi neural network. Eksperimen dilakukan dengan mengubah nilai epoch mulai dari 10-100. Kenaikan epoch antar eksperimen adalah 10. Hasil observasi dari eksperimen ditampilkan pada tabel

*Tabel 12: Nilai MAE terhadap Nilai Epoch*

Epoch	10	20	30	40	50	60	70	80	90	100
MAE	4.7339	3.7885	3.1633	2.9616	2.8771	2.8657	2.9134	2.8126	2.9285	2.6597

Berdasarkan hasil di atas, terlihat bahwa nilai MAE paling minimal terdapat pada kolom ke-6. Sehingga, untuk berikutnya, akan digunakan nilai 60 sebagai nilai epoch.

#### **Menentukan jumlah layer & unit**

Selanjutnya, adalah menentukan jumlah unit dari tiap hidden layer yang digunakan. Untuk memudahkan kombinasi, jumlah layer yang akan diuji mulai dari 2-10. Sedangkan untuk jumlah unit per layer yang digunakan adalah 32, 64, 128, dan 256. Belum dilakukan jumlah kombinasi unit per layer dalam sebuah arsitektur. Hyperparameter lain pada eksperimen ini adalah:

- 1) Nilai epoch: 60
- 2) Optimizer: RMSProp
- 3) Loss function: MSE
- 4) Activation function: ReLU

Berikut adalah hasil dari MAE berdasarkan kombinasi jumlah layer dan unit.

*Tabel 13: Eksperimen Jumlah Layer dan Jumlah Unit terhadap MAE*

Jum layer & unit	2@32	2@64	2@128	2@256	3@32	3@64	3@128	3@256
MAE	3.2293	2.8657	2.9181	2.9415	2.944	2.5892	2.9151	2.9562
Jum layer & unit	4@32	4@64	4@128	4@256	5@32	5@64	5@128	5@256
MAE	2.9546	2.9711	4.2466	2.6076	2.7477	2.6749	3.5756	2.5823
Jum layer & unit	6@32	6@64	6@128	6@256	7@32	7@64	7@128	7@256
MAE	3.1666	3.471	2.5752	2.4202	2.7608	2.7226	2.4252	2.5565
Jum layer & unit	8@32	8@64	8@128	8@256	9@32	9@64	9@128	9@256
MAE	2.8476	3.0723	2.6605	3.3307	2.5905	2.8046	2.4939	2.9707
Jum layer & unit	10@32	10@64	10@128	10@256				
MAE	3.4583	2.6132	3.5661	5.7842				

Eksperimen di atas menghasilkan 2.4202 sebagai nilai MAE terkecil. Nilai ini didapat dari jumlah hidden layer yang masing-masing layernya memiliki 256 unit. Selain eksperimen ini, dilakukan juga beberapa kombinasi jumlah unit untuk layer yang berbeda di 1 arsitektur yang sama. Akan tetapi, nilai yang dihasilkan tetap lebih besar dari 2.4202. Kombinasi jumlah & unit dari layer ini akan dijadikan acuan untuk eksperimen selanjutnya.

### Menentukan fungsi aktivasi

Fungsi aktivasi yang diuji untuk masing-masing kasus adalah Sigmoid, Tanh, Leaky ReLU, PReLU, ELU, dan ReLU. Konfigurasi untuk melakukan eksperimen ini sama dengan eksperimen sebelumnya. Hanya saja unit layer dan jumlah layer ditetapkan 256 dan 6.

*Tabel 14: Hasil Eksperimen Pengubahan Fungsi Aktivasi*

Fungsi Aktivasi	Sigmoid	Tanh	Leaky ReLU	PReLU	ELU	ReLU
MAE	6.5345	3.027	3.1614	2.6731	3.4222	2.4202

Hasil eksperimen memperlihatkan bahwa ReLU merupakan fungsi aktivasi yang menghasilkan nilai metric paling minimal.

### Menentukan optimizer

Untuk menentukan optimizer, hyperparameter yang ditetapkan sama dengan eksperimen sebelumnya, fungsi aktivasi ReLU, menggunakan jumlah layer 6, dengan masing-masing unit berjumlah 256. Terdapat 7 optimizer yang digunakan pada eksperimen ini. Tabel 15 memperlihatkan hasil percobaan yang dilakukan.

Fungsi Aktivasi	Adam	Adadelta	Adagrad	Adamax	NAdam	FTRL	RMSProp
MAE	3.2766	17.527	2.9931	2.6157	3.7052	3.4313	2.4202

Optimizer dengan nilai metric minimal didapat pada fungsi aktivasi RMSProp.

#### 1.2.2. Parameter yang Digunakan

Berdasarkan hasil observasi dari eksperimen yang dilakukan, berikut adalah parameter yang dapat digunakan untuk melakukan training terhadap data boston house pricing.

- a) Jumlah layer: 6
- b) Jumlah unit tiap layer: masing-masing 256
- c) Activation function: ReLU
- d) Optimizer: RMSProp