

# Laporan Tugas Eksplorasi Hyperparameter CNN dan Neural Network

## A. Persoalan Regresi

Pada ekplorasi Hyperparamter pada Fully Connected layer digunakan dataset *Boston House Price* yang terdiri dari 506 baris dan 14 kolom. Baris menandakan jumlah dataset yang akan digunakan dan kolom menandakan 13 variabel (fitur) dan satu variable target (MEDV). Sampel dataset *Boston House Price* ditunjukkan pada Tabel 1. Sebelum proses simulasi dataset di tidak memiliki nilai NaN yang bisa menyebabkan error.

Tabel 1. Sampel Dataset *Boston House Price*

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.90	5.33	36.2
5	0.02985	0.0	2.18	0	0.458	6.430	58.7	6.0622	3	222.0	18.7	394.12	5.21	28.7
6	0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311.0	15.2	395.60	12.43	22.9
7	0.14455	12.5	7.87	0	0.524	6.172	96.1	5.9505	5	311.0	15.2	396.90	19.15	27.1
8	0.21124	12.5	7.87	0	0.524	5.631	100.0	6.0821	5	311.0	15.2	386.63	29.93	16.5
9	0.17004	12.5	7.87	0	0.524	6.004	85.9	6.5921	5	311.0	15.2	386.71	17.10	18.9

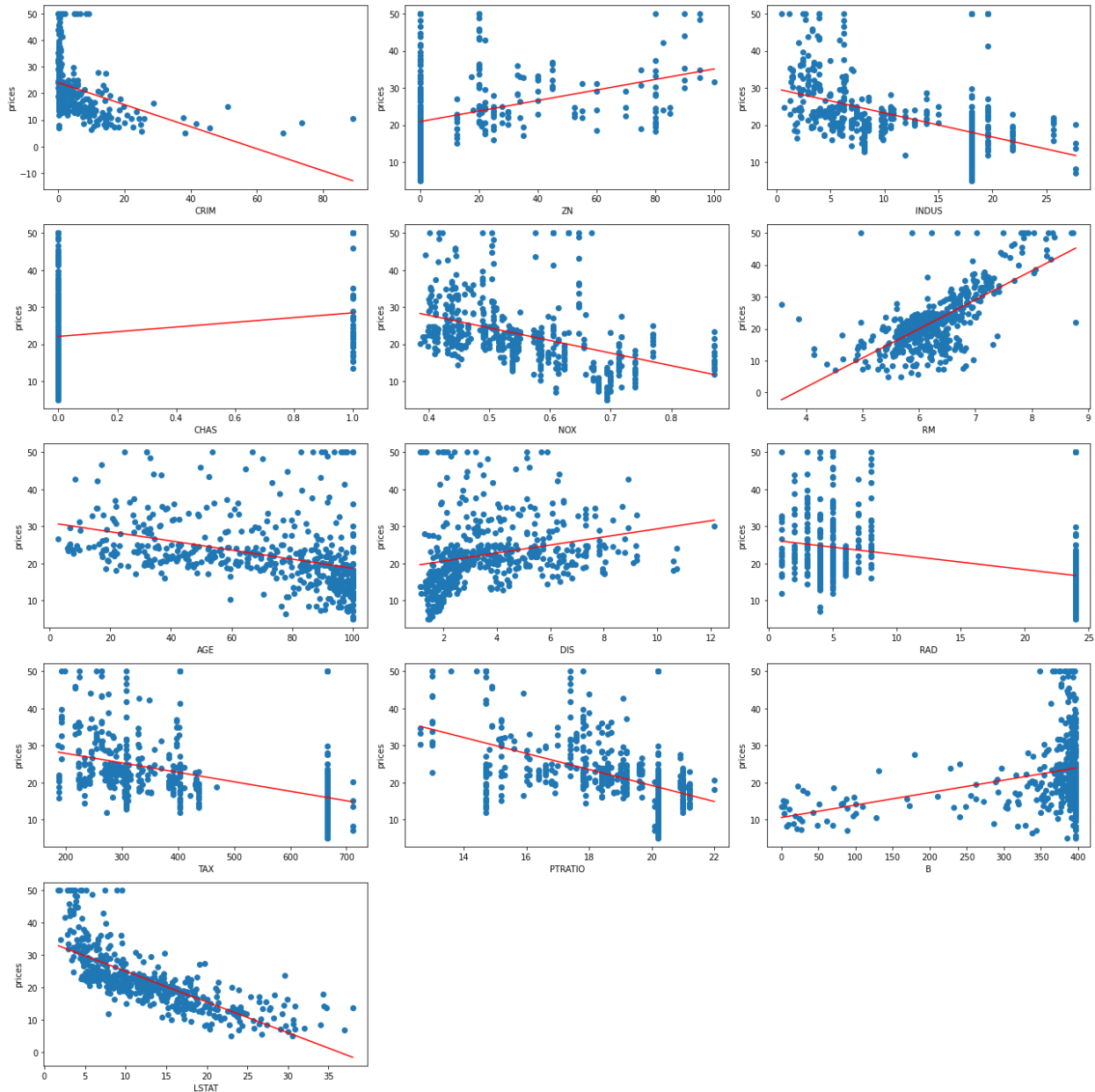
Untuk mengetahui bagaimana data dari setiap variabel tersebar dan mengetahui hubungannya dengan variabel target (MEDV) maka dibuat scatter plot seperti yang ditunjukkan pada Gambar 1. Dari *scatter plot* idibawah, bisa dikatakan bahwa fitur seperti LSTAT, RM, san PIRATIO memiliki efek yang besar baik itu secara positif atau negatif yang bisa mempengaruhi prediksi dari harga rumah.

Tabel 1. Nilai Hyperparameter awal

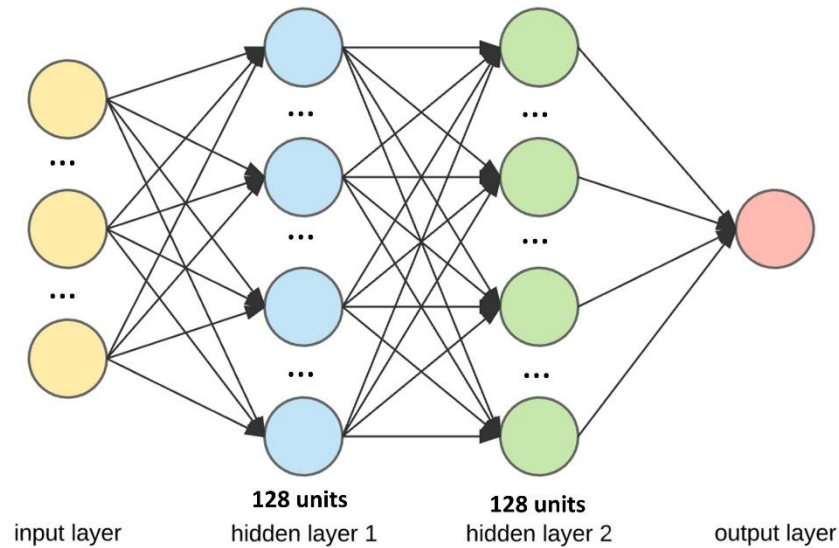
No	Hyperparameter	Nilai
1	Jumlah hidden units	128
2	Fungsi Aktivasi	ReLU
3	Jumlah Layer FC layer	2
4	Optimizer	Adam
5	Loss Function	Mean Squared Error
6	Epoch	300
7	Regularization	Early Stop

Selanjutnya untuk persiapan training, data dipisahkan antara variabel target dan fitur. Normalisasi diterapkan pada fitur karena rentang fitur tidak sama; beberapa fitur memiliki nilai maksimal 1 sedangkan beberapa memiliki 396. Lebih baik untuk menormalisasikan fitur sehingga model akan belajar lebih efisien.

Setelah itu, data dipecah untuk dataset pelatihan (80%) dan pengujian (20%). Proses mendapatkan model FC layer yang optimal dimulai dengan menetapkan arsitektur dan nilai *hyperparameter* awal yang akan digunakan. Arsitektur awal yang digunakan memanfaatkan 2 hidden layer dengan jumlah hidden units 128 untuk setiap layer nya. Hyperparameter lain yang digunakan dapat dilihat pada Tabel 1 dan untuk model arsitektur awalnya dapat dilihat pada Gambar 2.

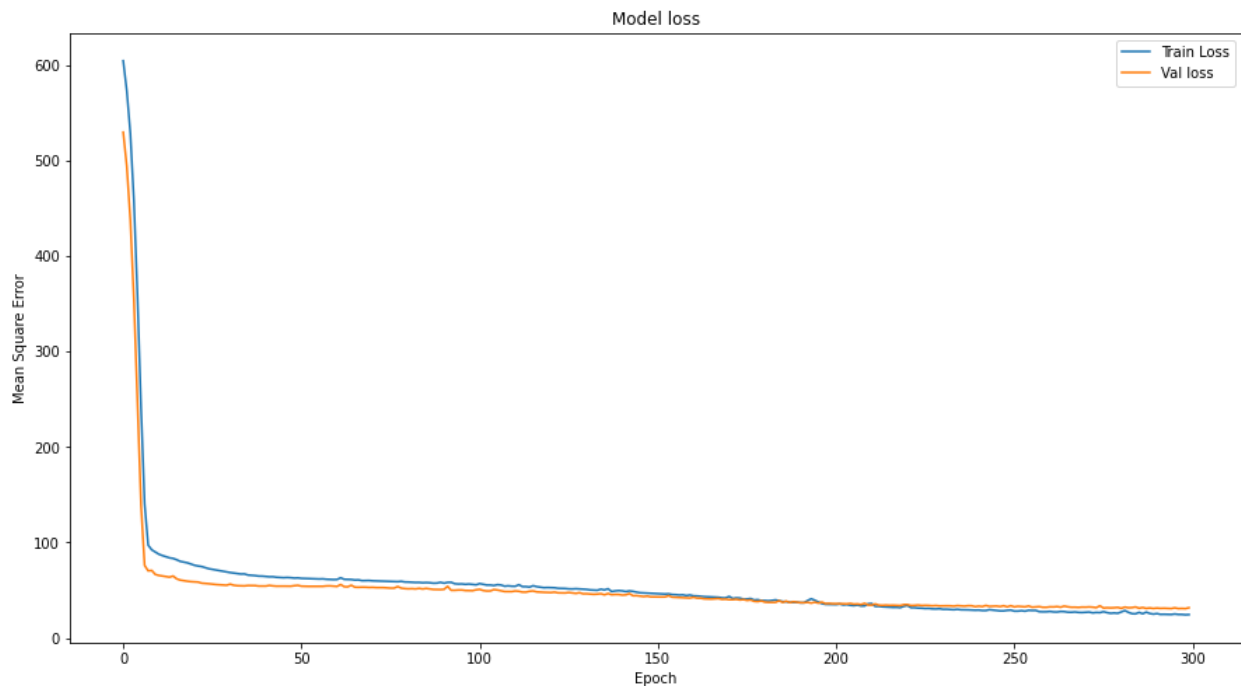


Gambar 1. Scatter plot 13 Variabel (fitur) terhadap Variabel Target (MEDV)



Gambar 2. Arsitektur awal untuk masalah regresi menggunakan dataset *Boston House Price*

Model dilatih menggunakan training set dengan jumlah epoch sebesar 300 dan metode regularisasi '*early stop*' diterapkan apabila model tidak mendapatkan peningkatan kinerja selama pelatihan, Early Stop akan menghentikan proses pelatihan agar tidak terjadi pengulangan yang akan mengurangi kinerja model yang diuji. Hasil evaluasi data uji dengan menggunakan arsitektur dan hyperparameter awal didapatkan nilai **mean absolute error** sebesar 3.377 dan loss sebesar 22.015. Gambar 3 menunjukkan grafik MSE untuk data pelatihan dan data validasi yang terus menurun dengan penambahan jumlah epoch-nya.

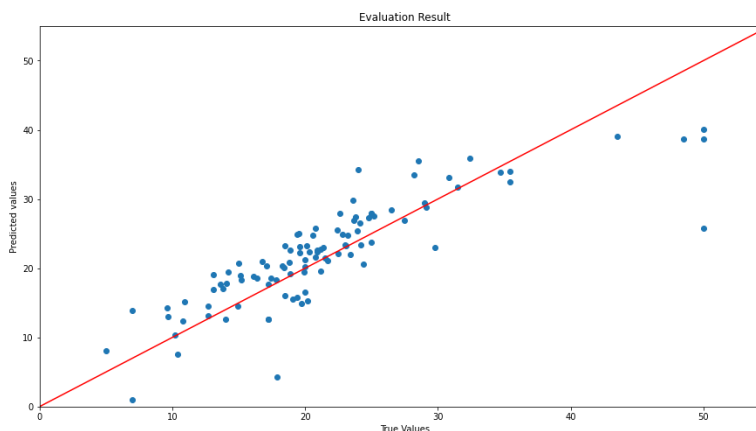


Gambar 3. Grafik MSE untuk data pelatihan dan validasi

Setelah titik epoch tertentu, nilai validasi dan training loss tetap hampir sama. Pada akhir grafik terlihat bahwa nilai loss validasi lebih besar dari nilai kerugian pelatihan. Hal ini benar karena validasi dilakukan pada kumpulan data baru. Untuk itu diperlukan uji validasi untuk melihat kemampuan model dalam mempelajari data baru. Tabel 2 menunjukkan hasil prediksi dan nilai sebenarnya pada pengujian menggunakan data Uji (test set).

Tabel 2. Hasil Nilai Prediksi dan Nilai Sebenarnya pada sampel Data Uji

	True Value	Predicted Value
0	23.6	29.815237
1	32.4	35.920536
2	13.6	17.709703
3	22.8	24.900223
4	16.1	18.802937
5	20.0	21.183872
6	17.8	18.320051
7	14.0	12.565032
8	19.6	23.063095
9	16.8	20.953629
10	21.5	21.495022
11	18.9	19.154602
12	7.0	0.981693
13	21.2	19.627125
14	18.5	23.238079



Gambar 4. Scatter Plot untuk hasil simulasi awal pada data uji

Gambar 4 menunjukkan *scatter plot* yang digambar antara nilai prediksi dan nilai sebenarnya menggunakan garis diagonal (45 derajat). Jadi, jika nilai prediksi hampir atau persis sama dengan nilai sebenarnya maka titik data jatuh pada garis diagonal 45 derajat. Sebagai contoh, jika nilai sebenarnya (sumbu x) adalah 25 dan nilai prediksi (sumbu y) adalah 25, maka titik koordinatnya adalah (25, 25) dan harus terletak pada

garis diagonal 45 derajat. Dari Gambar 4 tersebut, dapat dilihat bahwa sebagian besar titik data terletak dekat dengan garis merah. Dengan demikian, sebagian besar data uji diprediksi dengan sangat baik. Titik data yang jauh dari garis merah berarti selisih antara nilai prediksi dan nilai sebenarnya lebih besar. Titik data yang lebih dekat dengan garis merah menunjukkan lebih sedikit kesalahan dalam prediksi. Dengan kata lain, model memprediksi sangat baik atau mendekati nilai sebenarnya.

### A.1 Hasil Eksplorasi Jumlah Hidden layer yang Optimal

Proses pengujian menggunakan jumlah hidden layer sebagai berikut 2 (default), 4, 8, dan 16 dan nilai hyperparameter lain dibuat tetap. Pada Tabel 3, menunjukkan penambahan jumlah hidden layer berbanding lurus dengan penambahan jumlah parameter tetapi tidak selalu berbanding lurus dengan kinerja yang dihasilkan oleh model yang digunakan. Selain itu, hasil simulasi menunjukkan bahwa 4 hidden layer menghasilkan nilai MAE yang terkecil (2.849) dibandingkan dengan jumlah hidden layer lainnya.

Tabel 3. Hasil Eksplorasi Jumlah Hidden Layer yang Optimal

Jumlah Hidden Layer	Jumlah Parameter	Loss	Mean Absolute Error (MAE)
2 (default)	18.433	22.015	3.377
<b>4</b>	<b>51.457</b>	<b>16.809</b>	<b>2.849</b>
8	117.505	18.225	2.866
16	249.601	27.904	4.146

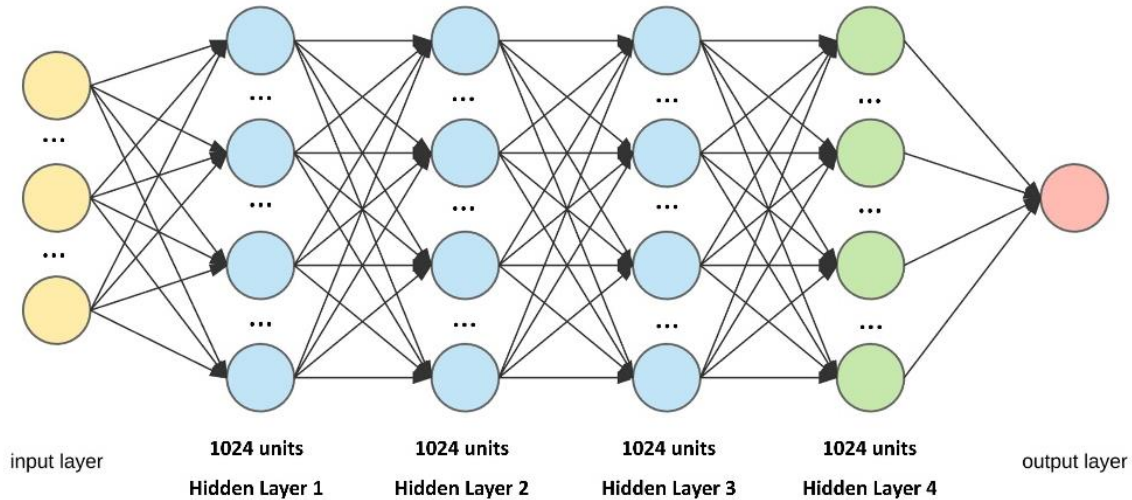
### A.2 Hasil Eksplorasi Jumlah Hidden Units yang Optimal

Melanjutkan dari simulasi A.1, arsitektur yang awalnya menggunakan 2 hidden layer digantikan dengan 4 hidden layer dimana pada proses simulasi setiap layer akan menggunakan jumlah hidden unit sebagai berikut; 64, 128 (default), 256, 512, dan 1024. Tabel 4 menunjukkan hasil simulasi jumlah hidden yang optimal untuk masalah regresi ini. Jumlah hidden units 1024 untuk setiap layer mendapatkan hasil terbaik dengan MAE sebesar 2.663 dan loss sebesar 18.029. Dari Tabel 4 juga diketahui bahwa dengan penambahan jumlah hidden units tidak selalu berbanding lurus dengan kinerja dari model yang digunakan, seperti pada jumlah hidden layer 64 mendapatkan MAE yang lebih baik dibanding jumlah hidden layer 256.

Tabel 4. Hasil Eksplorasi Jumlah Hidden Units yang Optimal

Jumlah Hidden Layer	Jumlah Parameter	Loss	Mean Absolute Error (MAE)
64	13.441	24.674	3.870
128 (default)	51.457	16.809	2.849
256	201.217	27.493	4.264
512	795.649	18.297	2.943
<b>1024</b>	<b>3.164.161</b>	<b>18.029</b>	<b>2.663</b>

Dari percobaan A.1 dan A.2 ini, terjadi perubahan arsitektur yang awalnya hanya terdiri dari 2 hidden layer dan masing-masing memiliki 128 hidden units menjadi 4 hidden layer dengan 1024 hidden unit untuk setiap layer-nya. Gambar 5 menunjukkan arsitektur yang akan digunakan selanjutnya berdasarkan hasil yang didapatkan sejauh ini.



Gambar 5. Perubahan Arsitektur berdasarkan percobaan A.1 dan A.2

### A.3 Hasil Eksplorasi Fungsi Aktivasi yang Optimal

Pada percobaan ini menggunakan beberapa fungsi aktivasi yang telah disediakan oleh keras, diantaranya: ReLU (*default*), Elu, GELU, tanh, dan sigmoid. Dari hasil simulasi, menunjukkan bahwa fungsi aktivasi ReLU menghasilkan MAE yang terkecil dibanding dengan aktivasi fungsi lainnya. Keuntungan utama menggunakan fungsi ReLU dibandingkan fungsi aktivasi lainnya adalah tidak mengaktifkan semua neuron secara bersamaan. Dalam praktiknya, model dengan menggunakan ReLU cenderung menunjukkan kinerja konvergensi yang lebih baik. Hasil simulasi untuk beberapa fungsi aktivasi ditunjukkan pada Tabel 5.

Tabel 5. Hasil Eksplorasi Fungsi Aktivasi yang Optimal

Fungsi Aktivasi	Loss	Mean Absolute Error (MAE)
<b>ReLU (<i>default</i>)</b>	<b>18.029</b>	<b>2.663</b>
ELU	34.429	4.841
GELU	25.143	3.811
tanh	25.659	3.787
sigmoid	74.636	6.207

### A.4 Hasil Eksplorasi Optimizer yang Optimal

Tabel 6. Hasil Eksplorasi Optimizer yang Optimal

Jumlah Hidden Layer	Loss	Mean Absolute Error
<b>Adam (<i>default</i>)</b>	<b>18.029</b>	<b>2.663</b>
SGD	17.953	2.829
RMSprop	36.692	4.182
Adadelta	55.599	5.413
Adagrad	51.629	4.613
Adamax	17.909	2.756
Nadam	30.979	4.659
Ftrl	46.363	4.653

Tabel 6 menunjukkan hasil eksplorasi dari optimizer yang disediakan oleh keras. Hasil simulasi menunjukkan bahwa optimizer adam memiliki kinerja yang lebih baik dibanding dengan optimizer lainnya dengan nilai MAE sebesar 2.663.

## A.5 Hasil Eksplorasi Fungsi Loss yang Optimal

Tabel 7. Hasil Eksplorasi Fungsi Loss yang Optimal

Jumlah Hidden Layer	Loss	Mean Absolute Error
Mean Squared Error ( <i>default</i> )	18.029	2.663
Mean Absolute Error	3.360	3.360
Mean Absolute Percentage Error	18.647	4.077
Mean Squared Logarithmic Error	0.040	2.830
Cosine Similarity	-1	21.267
<b>Huber</b>	<b>2.1521</b>	<b>2.601</b>

Untuk hasil yang didapatkan dengan menggunakan optimizer yang berbeda fungsi *Huber loss* mendapat kinerja yang terbaik dibanding optimizer lainnya dengan MAE sebesar 2.601, ditunjukkan pada Tabel 7. Dalam statistik, *loss Huber* adalah fungsi loss yang digunakan dalam regresi, yang kurang sensitif terhadap outlier dalam data daripada *squared error loss*. Dalam dataset *boston house* ini sendiri banyak outlier yang ada pada data, hal ini bisa dilihat pada Gambar 1. Outliers sendiri adalah data yang menyimpang terlalu jauh dari data lainnya yang membuat analisis terhadap serangkaian data menjadi bias, atau tidak mencerminkan fenomena yang sebenarnya. Sehingga hal ini yang menyebabkan *Huber loss* memiliki kinerja yang lebih baik dibanding dengan fungsi *loss* lainnya.

## Kesimpulan

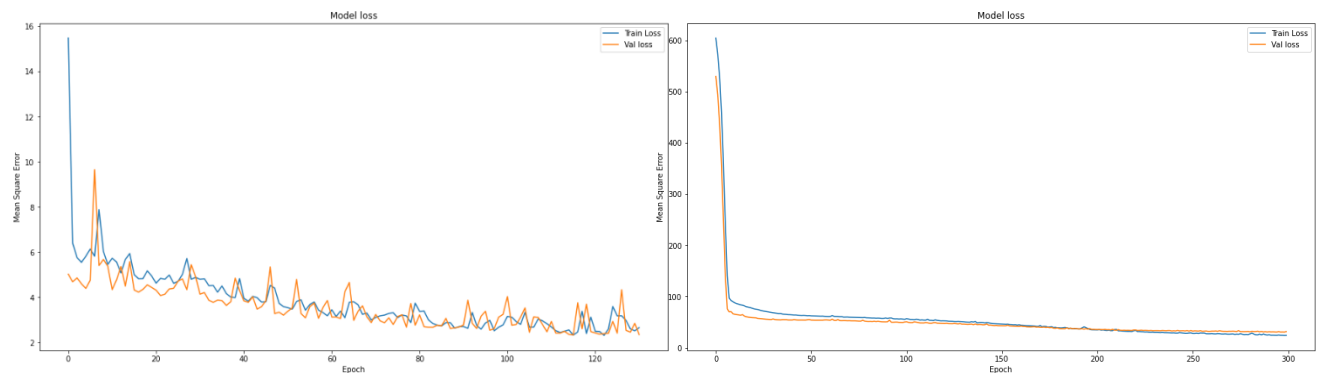
Dari hasil eksplorasi beberapa hyperparameter yang disimulasikan terdapat perubahan yang dilakukan dengan melihat hasil kinerjanya, diantara lain yaitu jumlah layer dari 2 menjadi 4 hidden layer dan jumlah hidden units dari 128 menjadi 1024. Selain itu, fungsi loss *Mean Squared Error* digantikan dengan Huber, Perubahan lebih lengkapnya dapat dilihat pada Tabel 8, dibawah ini :

Tabel 8. Nilai Hyperparameter akhir

No	Hyperparameter	Nilai
1	Jumlah hidden units	1024
2	Fungsi Aktivasi	ReLU
3	Jumlah Layer FC layer	4
4	Optimizer	Adam
5	Loss Function	Huber
6	Epoch	300
7	Regularization	Early Stop

Pada percobaan terakhir ini didapatkan nilai MAE pada data uji sebesar 2,601 dimana terdapat penurunan MAE sekitar 0.776 dari hasil percobaan awal yang dilakukan. Selain itu pada Gambar 6, terlihat perbandingan grafik *Model loss* untuk percobaan awal dan percobaan akhir dimana percobaan akhir

memiliki grafik yang lebih fluktuasi (huber loss) dibanding dengan percobaan awal (MSE loss). Namun, penggunaan teknik *early stop* memungkinkan model untuk mendapatkan kinerja yang lebih baik sebelum epochnya selesai. Hal ini menunjukkan bahwa proses *hyperparameter tuning* sangat dibutuhkan untuk mendapatkan model dengan kinerja yang optimal.



Gambar 6. Grafik akurasi (kanan) untuk percobaan awal dan (kiri) percobaan akhir

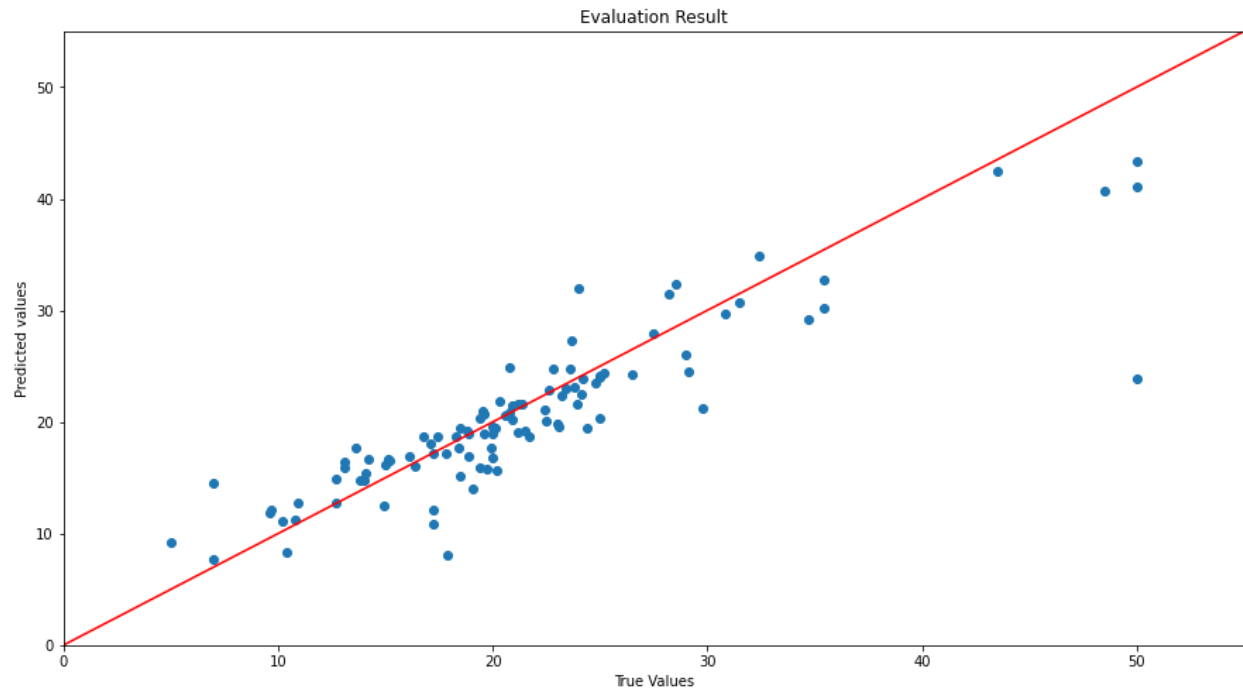
Tabel 9. Perbandingan Nilai Prediksi dan Nilai Sebenarnya pada sampel data uji untuk percobaan awal (kiri) dan akhir (kanan)

True Value Predicted Value			True Value Predicted Value		
0	23.6	29.815237	0	23.6	24.789328
1	32.4	35.920536	1	32.4	34.940899
2	13.6	17.709703	2	13.6	17.658688
3	22.8	24.900223	3	22.8	24.717955
4	16.1	18.802937	4	16.1	16.978939
5	20.0	21.183872	5	20.0	18.887434
6	17.8	18.320051	6	17.8	17.129154
7	14.0	12.565032	7	14.0	14.785150
8	19.6	23.063095	8	19.6	20.717381
9	16.8	20.953629	9	16.8	18.636587
10	21.5	21.495022	10	21.5	19.184467
11	18.9	19.154602	11	18.9	16.942631
12	7.0	0.981693	12	7.0	7.718163
13	21.2	19.627125	13	21.2	19.012068
14	18.5	23.238079	14	18.5	19.423265

Dari Tabel 8 menunjukkan bahwa percobaan akhir memiliki kinerja prediksi yang lebih baik dibanding percobaan awal. Sebagai contoh pada sampel no. 12 nilai yang diprediksi sangat jauh menyimpang dari



nilai sebenarnya (True value =7 dan predicted value=0.981693). Sebaliknya untuk percobaan akhir didapatkan hasil prediksi dan nilai sebenarnya hanya memiliki perbedaan sekitar 0.7. Gambar 7 menunjukkan secara keseluruhan sebaran data uji pada percobaan akhir ini.



Gambar 7. Scatter Plot untuk hasil simulasi akhir pada data uji