

Secara umum, eksplorasi hyperparameter yang dilakukan adalah sebagai berikut:

- Membuat model (CNN dan FC NN)
- Menentukan hyperparameter yang akan dieksplorasi

CNN	<ul style="list-style-type: none"> - Jumlah layer dan stride - Jumlah filter - Ukuran filter - Jumlah unit pada fully connected layer - Optimizer - Learning rate - Loss function
NN	<ul style="list-style-type: none"> - Jumlah layer - Jumlah hidden unit pada tiap layer - Activation function - Optimizer - Loss function

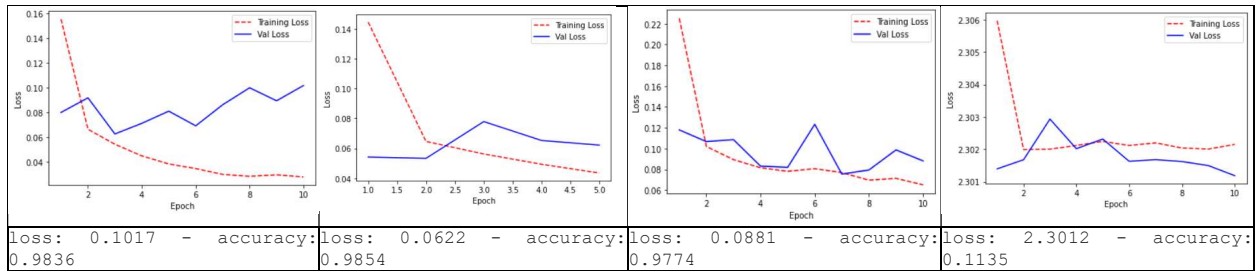
- Nilai hyperparameter dilakukan secara manual
- Hasil prediksi dari perubahan suatu hyperparameter dicatat dan dibandingkan satu sama lain dengan kriteria sebagai berikut:
 - Nilai loss relatif lebih rendah dibanding arsitektur lainnya.
 - Nilai loss mengalami penurunan hingga akhirnya relatif stabil.
 - Nilai loss antara training dan validation/test memiliki pola yang serupa dengan nilai loss training lebih rendah dibanding validation/test.
 - Tidak ada kenaikan loss pada validation/test yang signifikan seiring dengan penambahan epoch.
 - Jika ada hyperparameter yang grafiknya mirip, yang dianggap lebih baik adalah:
 - Naik turunnya nilai loss pada data training tidak terlalu tinggi
 - Arsitektur yang lebih sedikit membutuhkan resource

Eksplorasi CNN

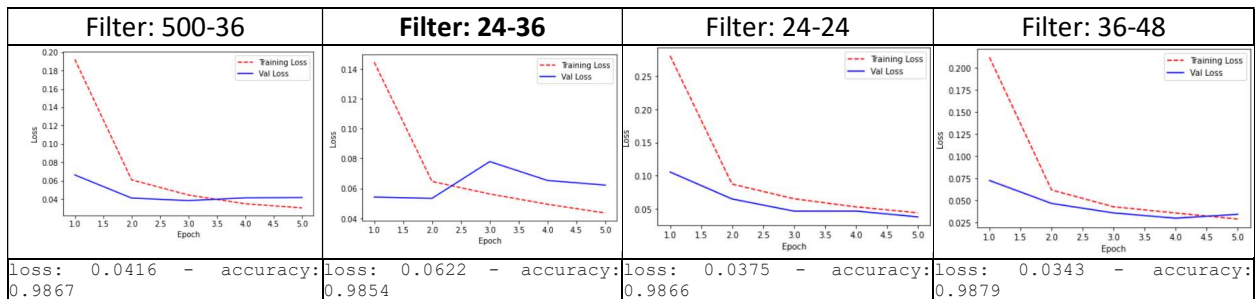
- Kinerja

Semakin banyak layer convolusi dan pooling yang digunakan nilai loss bertambah dan akurasi menurun, perbedaan loss antara data training dan data validasi juga semakin jauh. Sebaliknya jika menggunakan hanya satu layer convolusi dan pooling, mengakibatkan kenaikan loss pada data validasi. Dari eksplorasi jumlah layer diperoleh bahwa **2 layer** menggunakan **stride 2** memberikan kinerja paling baik dengan nilai loss 0.06 dan akurasi 0.98, grafik loss sempat mengalami kenaikan namun kembali menurun.

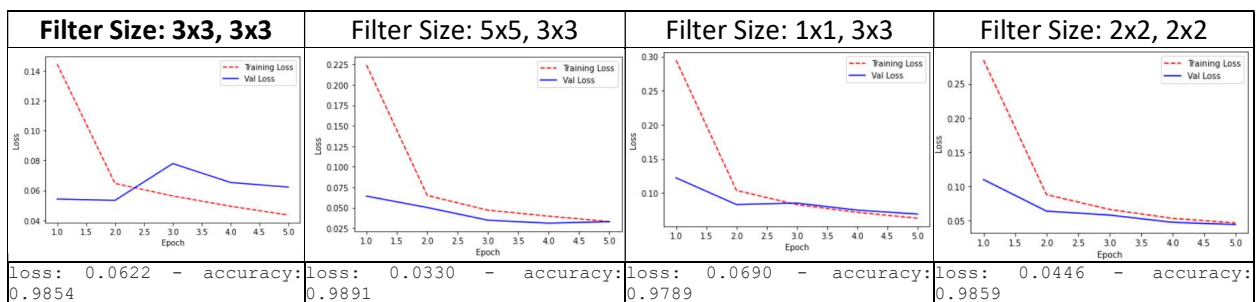
Layer: 1 Stride: 2 Filter: 24 -36 Filter Size: 3x3 Fc unit: 24	Layer: 2 Stride: 2 Filter: 24 -36 Filter Size: 3x3 Fc unit: 24	Layer: 3 Stride: 2 Filter: 24 -36 Filter Size: 3x3 Fc unit: 24	Layer: 3 Stride: 1 Filter: 24 -36 Filter Size: 3x3 Fc unit: 24
--	--	--	--



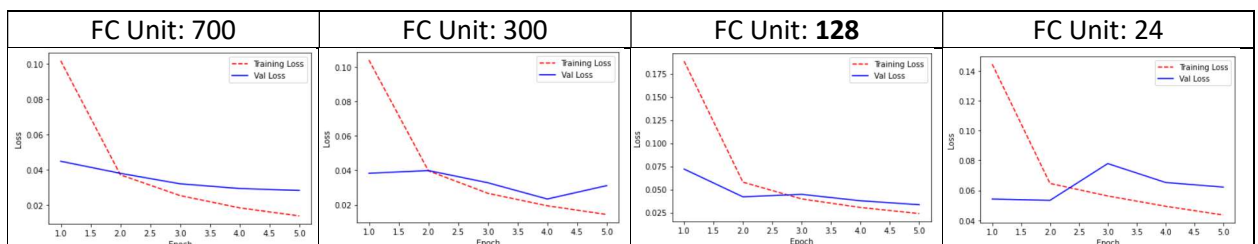
Dengan menggunakan 2 layer convolusi dan pooling selanjutnya dilakukan percobaan mencari jumlah filter yang memberikan kinerja yang baik. Selisih loss antara data training dan validasi paling kecil ditunjukkan jika model menggunakan filter sebanyak 24 pada layer convolusi pertama dan 36 pada layer convolusi berikutnya. Semakin besar jumlah filter semakin tinggi nilai loss yang didapat, selain itu jika jumlah filter pada layer sebelumnya sama atau lebih besar dibanding jumlah filter pada layer selanjutnya, nilai loss relatif stagnan.



Ukuran filter yang memberikan kinerja paling baik adalah 3 untuk masing-masing layer konvolusi. Percobaan untuk mencari ukuran filter menunjukkan semakin besar ukuran filter atau semakin kecil ukuran filter, nilai loss dataset validasi lebih rendah dibandingkan nilai loss dataset training.

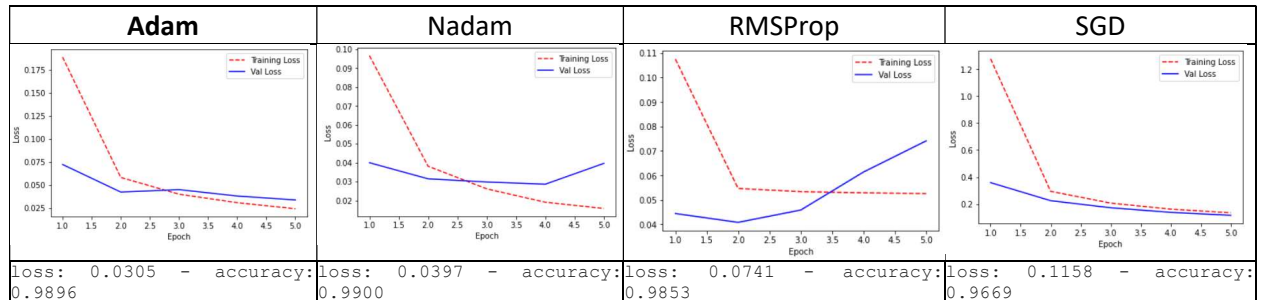


Hidden unit pada layer fully connected sejumlah 700 menghasilkan kinerja yang lebih baik dibanding 128. Karena perbedaannya tidak terlalu besar, dipilih jumlah hidden layer yang paling sedikit supaya waktu untuk menjalankan model relatif lebih cepat.

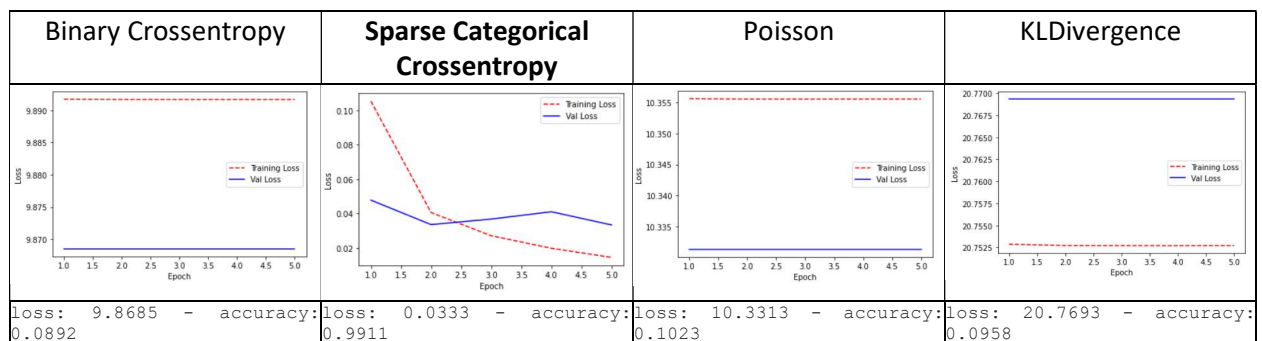


loss: 0.0284 - accuracy: 0.9921	loss: 0.0311 - accuracy: 0.9917	loss: 0.0305 - accuracy: 0.9896	loss: 0.0622 - accuracy: 0.9854
---------------------------------	---------------------------------	---------------------------------	---------------------------------

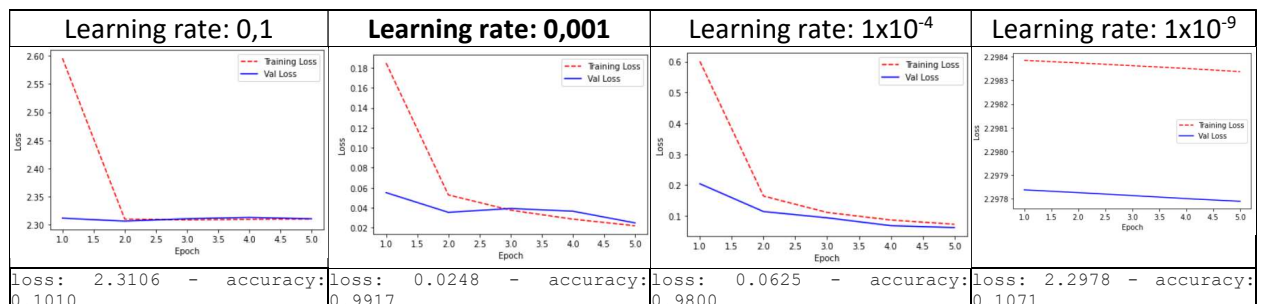
Seluruh optimizer yang disediakan Keras dicobakan terhadap model dan optimizer Adam menghasilkan kinerja yang paling baik. Optimizer lainnya menghasilkan grafik loss untuk dataset validasi yang lebih rendah dibandingkan grafik loss untuk dataset training. Optimizer yang tidak ditunjukkan pada gambar dibawah (Adadelta, Ftrl, Adagrad, Adamax) grafik nilai loss untuk validation dataset lebih rendah dibanding training dataset seperti halnya grafik loss SGD pada gambar dibawah ini. Optimizer Adadelta menghasilkan kinerja yang mirip dengan Ftrl sedangkan Adagrad dan Adamax menunjukkan kinerja yang mirip.



Loss function yang dicobakan untuk model hanya loss function untuk persoalan klasifikasi. Diantara berbagai loss function sparse categorical crossentropy memberikan kinerja yang baik. Sedangkan loss function lainnya lebih cocok digunakan untuk klasifikasi biner (hanya terdiri dari 2 kelas target)



Learning rate yang dicobakan pada eksplorasi ini dimulai dari 0.1 hingga 1×10^{-9} , learning rate yang memberikan kinerja paling baik adalah 0,001. Jika learning rate lebih besar dari 0,001 nilai loss relative stagnan, sedangkan learning rate yang lebih kecil mengakibatkan nilai loss pada validasi set lebih rendah dibanding nilai loss pada training set, semakin kecil learning rate, semakin jauh jarak antara nilai loss training set dan validation set



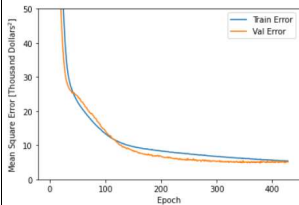
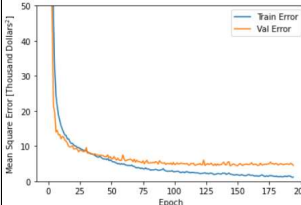
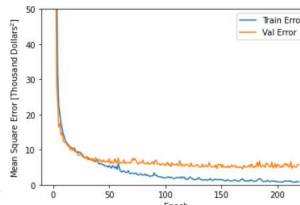
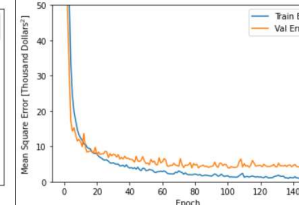
- Kesimpulan

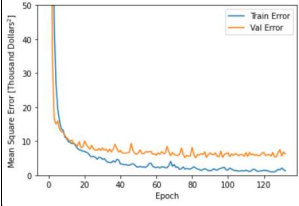
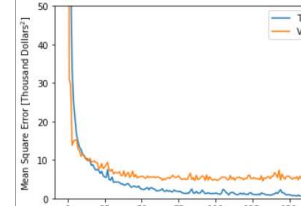
Dari eksplorasi yang dilakukan didapatkan arsitektur model terbaik untuk model CNN adalah menggunakan **2 layer** convolusi dan pooling dengan filter sebanyak **24 dan 36** masing-masing berukuran **3x3**, jumlah hidden unit pada fully connected layer adalah **128**. Model memberikan kinerja paling baik menggunakan optimizer **Adam**, dengan learning rate **0.001** dan loss function **sparse categorical crossentropy**.

Eksplorasi NN

- Kinerja

Jumlah layer yang dicoba untuk model fully connected neural network dimulai dari 2 layer hingga 9 layer dengan layer sejumlah 7 menghasilkan nilai loss paling rendah.

Layer: 2 Hidden Unit: 100 - 1 Activation: relu Optimizer: Adam Loss Function: mse	Layer:4 Hidden Unit: 3(100) - 1 Activation: relu Optimizer: Adam Loss Function: mse	Layer:5 Hidden Unit: 4(100)- 1 Activation: relu Optimizer: Adam Loss Function: mse	Layer:7 Hidden Unit: 6(100) - 1 Activation: relu Optimizer: Adam Loss Function: mse
			
RSME validation: 2.324 RSME Test: 3.81	RSME validation: 2.118 RSME Test: 3.415	RSME validation: 2.409 RSME Test: 3.27	RSME validation: 2.0 RSME Test: 3.136
loss: 13.3723 - mae: 2.4634 - mse: 13.3723	loss: 11.6612 - mae: 2.4405 - mse: 11.6612	loss: 10.6904 - mae: 2.2211 - mse: 10.6904	loss: 9.8327 - mae: 2.0615 - mse: 9.8327

Layer:8 Hidden Unit: 8(100)- 1 Activation: relu Optimizer: Adam Loss Function: mse	Layer:9 Hidden Unit: 9(100)- 1 Activation: relu Optimizer: Adam Loss Function: mse
	
RSME validation: 2.508 RSME Test: 3.2	RSME validation: 2.539 RSME Test: 3.755
loss: 10.2417 - mae: 2.2387 - mse: 10.2417	loss: 14.1027 - mae: 2.2867 - mse: 14.1027

Menggunakan 7 layer, selanjutnya dicoba beberapa jumlah hidden unit yang berbeda. Dari berbagai hidden unit yang dicoba, yang memiliki kinerja paling baik adalah sebanyak **100 unit**.

Hidden unit:5(200)-100-1	Hidden unit: 5(100)-20-1	Hidden unit:5(100)-50-1	Hidden unit:6(128)-1
RSME validation: 2.295 RSME Test: 3.369 loss: 11.3470 - mae: 2.0995 - mse: 11.3470	RSME validation: 2.33 RSME Test: 3.776 loss: 14.2563 - mae: 2.4097 - mse: 14.2563	RSME validation: 2.183 RSME Test: 4.003 loss: 16.0200 - mae: 2.4972 - mse: 16.0200	RSME validation: 1.983 RSME Test: 3.444 loss: 11.8595 - mae: 2.3591 - mse: 11.8595

Selanjutnya dilakukan percobaan untuk berbagai fungsi aktivasi relu, elu, sigmoid, tanh. Dari berbagai fungsi aktivasi yang digunakan, fungsi aktivasi **relu** memberikan nilai loss yang paling rendah selain itu, jika dilihat dari grafik loss antara fungsi aktivasi relu dan elu, menggunakan fungsi aktivasi relu naik turunnya nilai loss tidak terlalu tinggi.

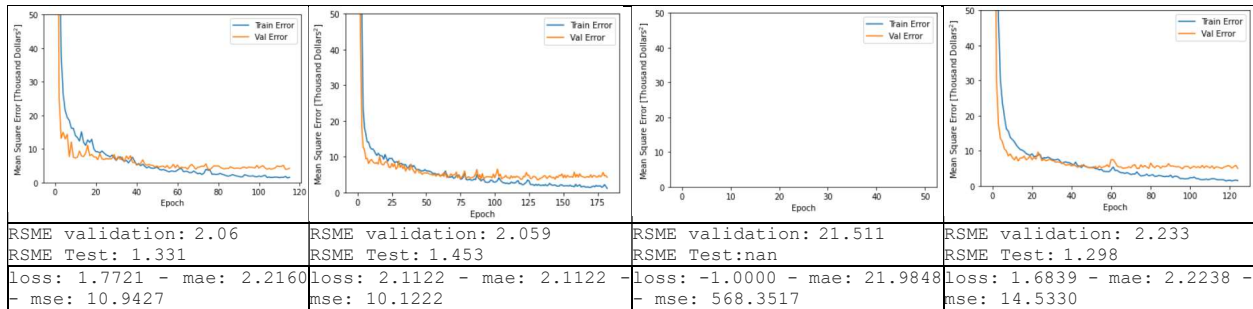
Activation: sigmoid	Activation:elu	Activation:tanh	Activation:Relu
RSME validation: 6.614 RSME Test: 9.139 loss: 83.5195 - mae: 6.5453 - mse: 83.5195	RSME validation: 2.224 RSME Test: 3.872 loss: 14.9955 - mae: 2.6568 - mse: 14.9955	RSME validation: 2.799 RSME Test: 4.453 loss: 19.8309 - mae: 2.8054 - mse: 19.8309	RSME validation: 2.423 RSME Test: 3.342 loss: 11.1678 - mae: 2.2994 - mse: 11.1678

Seluruh optimizer yang disediakan Keras dicobakan terhadap model dan optimizer **Adam** menghasilkan kinerja yang paling baik. Optimizer SGD sepertinya mengakibatkan terjadinya gradient explosion.

Optimizer:SGD	Optimizer:RMSProp	Optimizer:Ftrl	Optimizer:Adamax
RSME validation:nan RSME Test: nan loss: nan - mae: nan - mse: nan	RSME validation: 3.379 RSME Test: 4.377 loss: 19.1576 - mae: 3.3345 - mse: 19.1576	RSME validation: 2.568 RSME Test: 4.405 loss: 19.4080 - mae: 2.8032 - mse: 19.4080	RSME validation: 2.178 RSME Test: 3.618 loss: 13.0914 - mae: 2.3368 - mse: 13.0914

Loss function yang dicobakan untuk model hanya loss function untuk persoalan regresi, yakni: mse, mae, mape, msle, cosine_similarity, huber, dan logcosh. Loss function **huber** memberikan kinerja paling baik. Loss function MSE baik dalam mempelajari outlier yang ada pada data, sebaliknya MAE mengabaikan outlier tersebut. Loss function Huber menyeimbangkan MSE dan MAE.

Loss Function: Huber	Loss Function: MAE	Loss Function: cosine_similarity	Loss Function: logcosh
----------------------	--------------------	----------------------------------	------------------------



- Kesimpulan

Dari eksplorasi yang dilakukan didapatkan arsitektur model terbaik untuk model fully connected neural network adalah menggunakan **7 layer** dense dengan hidden unit pada masing-masing layer sejumlah **100** kecuali pada layer terakhir yang hanya terdiri dari 1 unit. Model memberikan kinerja paling baik menggunakan fungsi aktivasi **Relu**, optimizer **Adam**, dan loss function **Huber**.

Kesimpulan akhir

Pilihan hyperparameter pada satu area aplikasi belum tentu dapat digunakan pada area aplikasi lainnya. Ada banyak hyperparameter yang harus disesuaikan untuk mendapatkan kinerja model yang baik seperti jumlah layer, jumlah hidden unit, optimizer, dll. Selain hyperparameter yang dieksplorasi pada tugas ini masih ada hyperparameter lainnya yang sebaiknya juga disesuaikan seperti ukuran batch, jumlah epoch, momentum. Beberapa hyperparameter dapat dianggap lebih penting dibanding hyperparameter lainnya. Pada eksplorasi diatas dapat dilihat perubahan kecil pada learning rate berdampak besar terhadap kinerja model dibandingkan dengan perubahan jumlah layer atau jumlah hidden unit.

Beberapa hyperparameter harus dipilih berdasarkan domain permasalahan misalnya activation function dan loss function. Menggunakan loss function MSE untuk permasalahan klasifikasi misalnya tentu saja tidak akan memberikan kinerja yang baik.

Dalam melakukan eksplorasi hyperparameter, untuk menentukan nilai yang dicobakan sebaiknya menggunakan strategi tertentu untuk menghemat waktu dalam melakukan percobaan misalnya menggunakan random sampling atau grid search dan jika menemukan nilai yang cukup baik, focus ke nilai-nilai yang berada disekitar nilai tersebut.