

IMPLEMENTASI SUPERVISED LEARNING MENGGUNAKAN ARTIFICIAL NEURAL NETWORK UNTUK MEMPREDIKSI KEKUATAN BETON

oleh:

I Putu Sri Randha Yoga¹; Annisa Izzatul Latifa²; Rifqah Amaliyah³
prandhayoga@student.telkomuniversity.ac.id¹; annisaizzatull@student.telkomuniversity.ac.id²;
rifqahamaliyah@student.telkomuniversity.ac.id³

ABSTRAK

Beton merupakan bahan yang terbuat dari campuran air, semen, dan beberapa agregat lainnya, dan beton merupakan bahan yang memiliki banyak manfaat dalam membangun berbagai macam infrastruktur. Oleh karena itu, beton patut untuk diperhitungkan ketahanannya. Metode yang akurat dalam memprediksi kuat tekan beton dapat memberikan keuntungan yang signifikan terhadap industri material konstruksi. Penelitian ini mencoba memanfaatkan kecerdasan buatan untuk memetakan hubungan yang kompleks dalam campuran beton dengan tingkat akurasi yang tinggi. Empat model diuji untuk mengevaluasi prediksi kekuatan beton. Analisis model menunjukkan perbedaan performa akibat konfigurasi yang berbeda. Model baseline menunjukkan kinerja optimal berkat keseimbangan parameter. Model 1 dengan *epoch* lebih besar rentan terhadap *overfitting*. Model 2, dengan *learning rate* lebih kecil, mungkin mengalami ketidakseimbangan pada *epoch*. Sementara model 3, dengan lebih banyak lapisan dan neuron, memerlukan manajemen lebih baik untuk menghindari *overfitting*.

Kata kunci : Jaringan saraf tiruan, kecerdasan buatan, prediksi, kekuatan beton.

1. PENDAHULUAN

Beton merupakan material konstruksi yang umum digunakan di seluruh dunia karena memiliki karakteristik kekuatan yang stabil. Selain empat bahan dasar utama, yaitu semen, agregat kasar, agregat halus, dan air, berbagai bahan tambahan seperti *fly ash*, *blast furnace slag*, dan *superplasticizer* juga digunakan [1]. Penambahan bahan-bahan ini bermanfaat dari segi ekonomi karena dapat mengurangi penggunaan semen Portland, yang merupakan bahan paling mahal dalam campuran beton. Meski demikian, penggunaan bahan tambahan ini juga dapat memiliki dampak pada ketahanan dan kekuatan beton.

Pengujian kekuatan beton biasanya dilakukan 7 hingga 28 hari setelah pengecoran, hal ini tentunya menghambat proses kerja proyek konstruksi. Prediksi yang cepat dan akurat

mengenai kekuatan beton sangat penting untuk pengendalian kualitas, dan ini sangat menguntungkan jika dapat diperoleh sejak tahap perencanaan. Karena hubungan antara komponen beton bersifat non-linear, pemodelan dengan pendekatan matematis menjadi kompleks. Sejumlah besar penelitian telah dilakukan dengan menggunakan pengujian eksperimental yang biasanya memakan banyak waktu dan biaya. Oleh karena itu, diperlukan sistem pemodelan baru yang tidak bergantung pada eksperimen namun tetap mampu memprediksi kekuatan beton dengan akurasi yang baik.

Di era ini, metode kecerdasan buatan banyak digunakan untuk melakukan prediksi. Dalam upaya meningkatkan akurasi prediksi, *Artificial Neural Network* (ANN) hadir sebagai sebuah model matematika yang dikomputasi untuk meniru struktur jaringan saraf otak manusia [2]. ANN

terdiri dari *artificial neuron* yang berkomunikasi satu sama lain. Dalam konteks regresi, ANN dapat membantu untuk memahami serta menggambarkan hubungan yang rumit antara berbagai variabel *input* dan *output*.

Pada penelitian ini akan dicoba penerapan regresi ANN pada *dataset Concrete Compressive Strength* dari UCI Machine Learning Repository untuk membuat model prediksi yang mampu mengestimasi kekuatan beton pada berbagai kombinasi komposisi material. Hal ini memungkinkan insinyur sipil, arsitek, dan perancang struktur untuk mengambil keputusan yang lebih baik dalam perancangan struktur beton yang kuat dan tahan lama, berdasarkan pemahaman yang mendalam tentang faktor-faktor yang mempengaruhi kekuatan tekan beton.

1. METODE PENELITIAN

1.1. Pemilihan Dataset

Pada penelitian ini, *dataset* yang digunakan adalah *dataset Concrete Compressive Strength* dari UCI Machine Learning Repository untuk membuat model prediksi yang mampu mengestimasi kekuatan beton pada berbagai kombinasi komposisi campuran material. Dataset ini terdiri dari 1030 baris data dengan 9 atribut yaitu :

Atribut Name	Data Type
Cement	float64
Blast Furnace Slag	float64
Fly Ash	float64
Water	float64
Superplasticizer	float64
Coarse Aggregate	float64
Fine Aggregate	float64
Age	int
Concrete	float64

Compressive Strength	
----------------------	--

Tabel 1.1 Atribut dataset

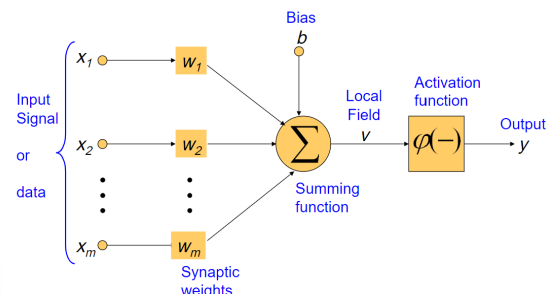
1.2. Pemrosesan dan Visualisasi Data

Pada tahapan ini, dilakukan langkah-langkah pra pemrosesan data, yang mencakup *data cleaning*, mengidentifikasi *missing value*, *data duplicate*, serta *outlier* pada data. Sementara itu, tahap visualisasi data bertujuan untuk mendalami pemahaman mengenai bagaimana seluruh atribut berhubungan dengan atribut target.

1.3. Pemilihan Model

Artificial Neural Network atau dalam bahasa Indonesia disebut Jaringan Saraf Tiruan adalah model komputasi yang meniru struktur dan aspek fungsional dari sistem saraf manusia. Aplikasi Jaringan Saraf Tiruan terbagi atas klasifikasi dan regresi. Neuron adalah sebuah unit pemroses informasi yang menjadi dasar pengoperasian jaringan saraf tiruan. Saraf adalah sebuah unit pemroses informasi dengan tiga elemen dasar yaitu :

- 1) Satu set *link* yang terhubung
- 2) Sebuah penjumlah untuk menghitung besarnya penambahan pada sinyal masukan
- 3) Sebuah fungsi aktivasi untuk membatasi banyaknya keluaran pada saraf



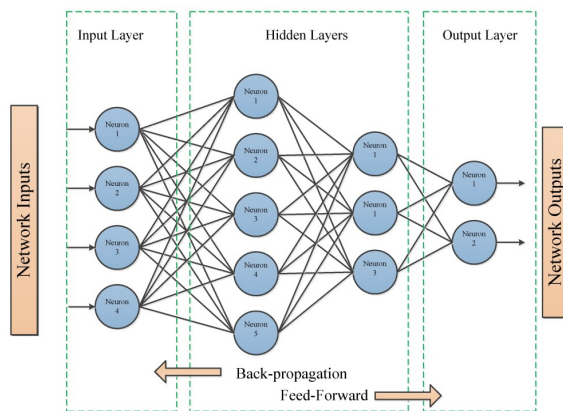
Gambar 2.1 Neural Network

Model Jaringan Saraf Tiruan yang umum digunakan adalah *multilayer perceptron* (MLP).

Dalam model MLP, lapisan *input* terdiri dari node-node input sensorik, satu atau lebih *hidden* yang berfungsi untuk perhitungan, dan lapisan output memuat hasil komputasi. Algoritma yang banyak digunakan dan efektif untuk melatih model MLP adalah algoritma *backpropagation* (BP), yang sering digunakan dalam prediksi non-linear [3].

Pada dasarnya, pelatihan dengan metode *backpropagation* terdiri atas tiga langkah, yaitu :

- 1) Data dimasukkan ke input jaringan (*feed forward*)
- 2) Perhitungan dan propagasi balik dari *error* yang bersangkutan
- 3) Pembaharuan (*adjustment*) bobot dan bias.



Gambar 2.2 Multilayer preceptron

1.4. Pembagian data

Untuk menguji serta mengevaluasi performa model, dataset dipecah menjadi dua kelompok utama, yaitu data latih sebanyak 80.% dari dataset dan data uji sebanyak 20% dari dataset. Variabel *input* yang digunakan dalam penelitian ini mencakup atribut-atribut seperti *Cement*, *Blast Furnace Slag*, *Fly Ash*, *Water*, *Superplasticizer*, *Coarse Aggregate*, *Fine Aggregate*, dan *Age*. Sementara itu, atribut target yang ingin diprediksi adalah *Concrete Compressive Strength*.

1.5. Forward Propagation

Pada tahap ini, data input akan dikalikan dengan matriks bobot lalu ditambahkan dengan

vektor bias pada setiap lapisan jaringan. Untuk memperoleh non-linearitas, hasil dari perhitungan tersebut diteruskan ke fungsi aktivasi. Proses ini dilakukan secara berulang dari satu lapisan ke lapisan lainnya, dengan setiap lapisan mengalami transformasi melalui fungsi aktivasi. *Forward propagation* bertanggung jawab menghasilkan prediksi atau output dari jaringan saraf berdasarkan input yang diberikan.

1.6. Fungsi Aktivasi

ReLU (*Rectified Linear Unit*). Fungsi ini memperoleh hasil berdasarkan pemilihan nilai maksimum antara 0 dan nilai inputnya, yang mewakili karakteristik non-linearitas yang kuat dalam jaringan saraf. Adapun rumus aktivasi ReLU adalah sebagai berikut :

$$f(Z) = \max(0, Z).$$

Nilai turunan :

$$df(Z)/dZ = 1, Z > 0$$

$$df(Z)/dZ = 0, Z \leq 0.$$

Nilai turunan pada fungsi aktivasi penting dalam membantu menghitung seberapa besar perubahan yang dibutuhkan dalam model sesuai dengan perubahan yang terjadi dalam input *Z*.

1.7. Membuat Cost Function

Cost Function akan mengukur seberapa baik parameter model bekerja terhadap set pelatihan. Yang bertujuan untuk meminimalkan jumlah kuadrat terkecil, yang disebut metode *Ordinary Least Squares* (OLS). Cost function menghitung seberapa dekat hasil prediksi ($\hat{y}=AL$) dengan nilai sebenarnya (*y*). Hal ini dilakukan dengan mengevaluasi perbedaan kuadrat antara keduanya dan kemudian menormalisasi hasilnya.

1.8. Backward Propagation

Pada proses *backward propagation*, gradien fungsi *cost/loss* terhadap bobot diiterasi mulai dari *layer* terakhir. Tahapan ini melibatkan perhitungan

linear backward untuk menghitung *gradien* bobot dan bias, *linear activation backward* yang mengatur *gradien* fungsi aktivasi, serta *L-model-backward* yang menggabungkan langkah-langkah *backpropagation* untuk menghitung *gradien* di setiap lapisan jaringan dan mengubah bobot jaringan agar sesuai dengan data latih.

1.9. Training Model

Pada bagian ini, model *neural network* dilatih menggunakan metode *forward* dan *backward propagation*. Langkah-langkah ini terjadi iteratif sebanyak n kali, di mana setiap iterasi melibatkan perhitungan prediksi (*forward propagation*), perhitungan nilai biaya (*cost*), dan pembaruan parameter (*backward propagation*) berdasarkan *gradien cost*. Setiap iterasi juga mencatat nilai biaya dan kemudian menampilkan grafik hasil pelatihan untuk memantau perubahan nilai biaya seiring waktu.

1.10. Prediksi dan Evaluasi

Pada tahap ini, model dilatih menggunakan data latih (*training set*), lalu diuji menggunakan data uji (*test set*). Evaluasi kinerja model dilakukan dengan menggunakan *Mean Squared Error* (MSE), di mana perbandingan dilakukan antara prediksi model (*predictions*) dan nilai sebenarnya (*true observations*) dari sampel uji. Perhitungan MSE dilakukan untuk data latih (*MSE_{train}*) dan data uji (*MSE_{test}*) untuk mengevaluasi seberapa baik model memprediksi nilai sebenarnya. Semakin kecil nilai MSE, semakin baik kualitas prediksi model.

2. HASIL DAN PEMBAHASAN

2.1. Pra-pemrosesan Data

Tahap pra-pemrosesan data adalah langkah awal dalam pengolahan data sebelum data tersebut dapat digunakan dalam analisis atau pemodelan. Tujuan dari tahap pra-pemrosesan data adalah untuk membersihkan, mengorganisasi, dan mengubah data agar sesuai dengan kebutuhan

analisis atau model yang akan diterapkan. Beberapa tahap utama dalam pra-pemrosesan data meliputi:

1) Pengecekan *missing value*

```
RangeIndex: 1030 entries, 0 to 1029
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   Cement                 1030 non-null   float64
1   BlastFurnaceSlag       1030 non-null   float64
2   FlyAsh                 1030 non-null   float64
3   Water                  1030 non-null   float64
4   Superplasticizer       1030 non-null   float64
5   CoarseAggregate        1030 non-null   float64
6   FineAggregate          1030 non-null   float64
7   Age                    1030 non-null   int64  
8   CompressiveStrength    1030 non-null   float64
dtypes: float64(8), int64(1)
memory usage: 72.5 KB
```

Gambar 2.3 Dataset info

Dataset ini tidak mengandung data yang hilang, seperti yang ditunjukkan oleh gambar 2.3 bahwa semua atribut memiliki jumlah entri non-null sebanyak 1030, sesuai dengan jumlah total baris data.

2) Pengecekan data duplikat

```
Total Duplikasi: 25
Data Duplikasi:
Cement BlastFurnaceSlag FlyAsh Water Superplasticizer CoarseAggregate FineAggregate Age CompressiveStrength
77  425.0      106.3      0.0  153.5      16.5      852.1      887.1      3      33.398217
88  425.0      106.3      0.0  153.5      16.5      852.1      887.1      3      33.398217
88  362.6      189.0      0.0  164.9      11.6      944.7      755.8      3      35.381171
91  362.6      189.0      0.0  164.9      11.6      944.7      755.8      3      35.381171
100  425.0      106.3      0.0  153.5      16.5      852.1      887.1      7      49.281807
103  425.0      106.3      0.0  153.5      16.5      852.1      887.1      7      49.281807
109  362.6      189.0      0.0  164.9      11.6      944.7      755.8      7      55.895819
111  362.6      189.0      0.0  164.9      11.6      944.7      755.8      7      55.895819
123  425.0      106.3      0.0  153.5      16.5      852.1      887.1      28      60.294676
126  425.0      106.3      0.0  153.5      16.5      852.1      887.1      28      60.294676
132  362.6      189.0      0.0  164.9      11.6      944.7      755.8      28      71.298713
134  362.6      189.0      0.0  164.9      11.6      944.7      755.8      28      71.298713
137  362.6      189.0      0.0  164.9      11.6      944.7      755.8      28      71.298713
146  425.0      106.3      0.0  153.5      16.5      852.1      887.1      56      64.308532
149  425.0      106.3      0.0  153.5      16.5      852.1      887.1      56      64.308532
155  362.6      189.0      0.0  164.9      11.6      944.7      755.8      56      77.297154
157  362.6      189.0      0.0  164.9      11.6      944.7      755.8      56      77.297154
160  362.6      189.0      0.0  164.9      11.6      944.7      755.8      56      77.297154
169  425.0      106.3      0.0  153.5      16.5      852.1      887.1      91      65.196851
172  425.0      106.3      0.0  153.5      16.5      852.1      887.1      91      65.196851
177  362.6      189.0      0.0  164.9      11.6      944.7      755.8      91      79.296635
179  362.6      189.0      0.0  164.9      11.6      944.7      755.8      91      79.296635
182  362.6      189.0      0.0  164.9      11.6      944.7      755.8      91      79.296635
889  252.0      0.0      0.0  185.0      0.0      1111.0      784.0      28      19.691435
```

Gambar 2.4 Daftar data duplikat

Gambar 2.4 menunjukkan bahwa terdapat 25 baris data yang memiliki nilai yang identik untuk seluruh atributnya. Untuk menjaga integritas dan kualitas dataset, data duplikat tersebut kemudian akan dihapus.

3) Pengecekan *outlier*

```
➡ Jumlah Outlier per Atribut:  
  
Cement : 0 data outlier  
BlastFurnaceSlag : 2 data outlier  
FlyAsh : 0 data outlier  
Water : 15 data outlier  
Superplasticizer : 10 data outlier  
CoarseAggregate : 0 data outlier  
FineAggregate : 5 data outlier  
Age : 59 data outlier  
CompressiveStrength : 8 data outlier
```

Gambar 2.5 Jumlah Outlier pada Dataset

Gambar 2.5 memberikan informasi mengenai atribut-atribut yang mengandung outlier, yaitu data yang berada di luar rentang normal atau ekstrem. Dalam hal ini, atribut *Age* menunjukkan adanya jumlah outlier yang paling signifikan, dengan sebanyak 59 data *outlier*. Meski demikian, jika dibandingkan dengan jumlah total baris data sebanyak 1005, jumlah outlier ini relatif kecil. Oleh karena itu, diputuskan untuk tidak melakukan proses khusus terkait data outlier ini.

2.2. Eksplorasi Data

Pada tahap ini merupakan tahap awal dalam analisis data, yaitu proses menyelami dan memahami data untuk selanjutnya dianalisis sehingga mendapatkan pola, tren, dan *insight* yang berguna.

1) Cek statistik deskriptif dataset

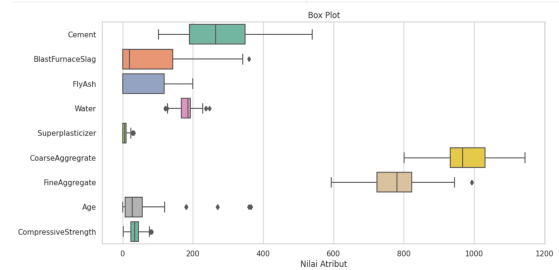
	Cement	BlastFurnaceSlag	FlyAsh	Water	Superplasticizer	CoarseAggregate	FineAggregate	Age	CompressiveStrength
count	1005.000000	1005.000000	1005.000000	1005.000000	1005.000000	1005.000000	1005.000000	1005.000000	1005.000000
mean	278.629055	72.043134	55.535075	182.074378	6.031647	874.378468	772.686617	45.856716	35.250273
std	104.345003	86.170555	64.207448	21.340740	5.919559	77.579534	80.338851	63.734892	16.284808
min	102.000000	0.000000	0.000000	121.750000	0.000000	801.000000	594.000000	1.000000	2.331898
25%	190.680000	0.000000	0.000000	166.610000	0.000000	932.000000	724.300000	7.000000	23.523542
50%	265.000000	20.000000	0.000000	185.700000	6.100000	968.000000	780.000000	28.000000	33.798114
75%	349.000000	142.500000	118.270000	192.940000	10.000000	1031.000000	822.200000	56.000000	44.968340
max	540.000000	359.400000	200.100000	247.000000	32.200000	1145.000000	992.600000	365.000000	62.599225

Gambar 2.6 Statistik deskriptif dataset

Gambar 2.6 menunjukkan informasi statistik deskriptif atribut dataset yang terdiri dari jumlah pengamatan dari dataset (*count*), nilai rata-rata setiap kolom (*mean*), simpangan baku (*std*), nilai terkecil dalam setiap atribut (*min*), kuartil pertama dengan membagi data menjadi 25% dibawah, kuartil kedua dengan membagi data

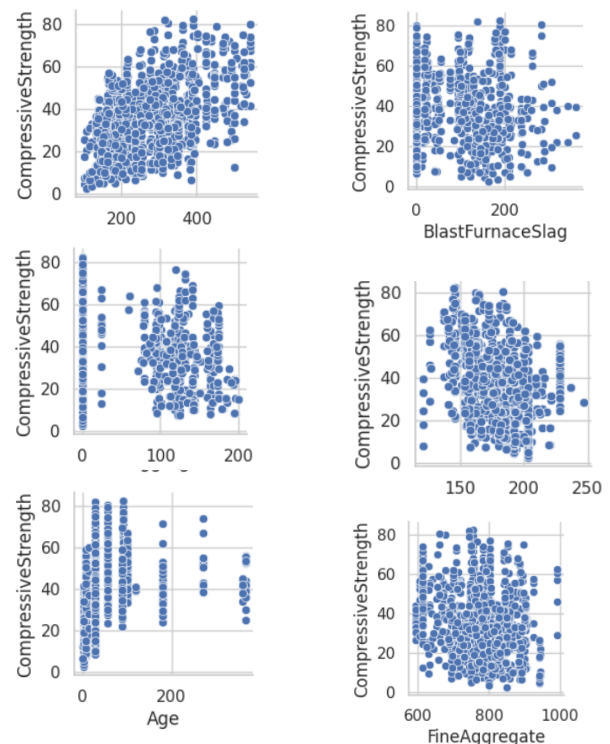
menjadi 50% dibawah, dan nilai terbesar dari setiap atribut (*max*).

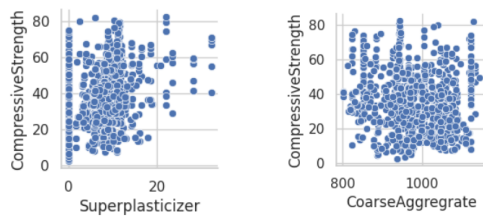
2) Visualisasi seluruh atribut dengan *boxplot*



Dari gambar 2.7 dapat dilihat bahwa atribut *Age* memiliki *outlier* paling banyak diantara atribut lainnya.

3) Visualisasi *pairplot* untuk melihat korelasi antar atribut terhadap *Compressive Strength*

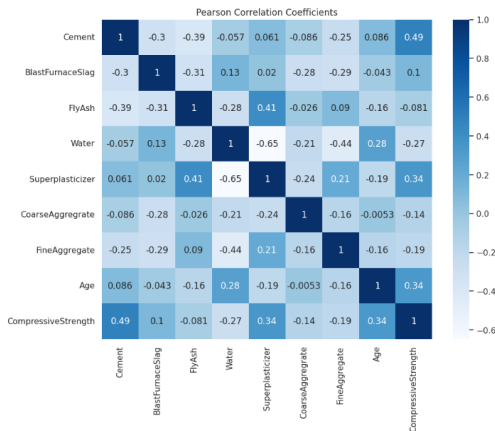




Gambar 2.8 Scatter plot seluruh atribut terhadap atribut Compressive Strength

Berdasarkan gambar 2.8 korelasi atribut yang konstan terdapat pada atribut *Cement* dimana kumpulan dari pairplot tersebut naik secara konstan.

4) Visualisasi *pearson correlation coefficient*



Gambar 2.9 Pearson correlation coefficient

Gambar 2.9, menunjukkan bahwa atribut *Cement* memiliki korelasi positif tertinggi terhadap *Compressive Strength*, dengan nilai sebesar 0.49, diikuti oleh *Age* dan *Superplasticizer*, dengan korelasi sebesar 0.34.

5) Mengurutkan nilai-nilai korelasi pearson antar pasangan atribut

Head		

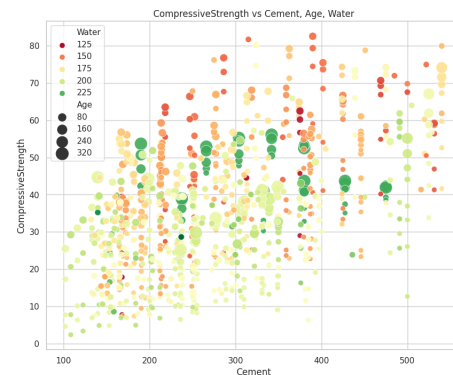
CompressiveStrength	Cement	0.488283
Cement	CompressiveStrength	0.488283
Superplasticizer	FlyAsh	0.414055
FlyAsh	Superplasticizer	0.414055
CompressiveStrength	Superplasticizer	0.344225
Superplasticizer	CompressiveStrength	0.344225
Age	CompressiveStrength	0.337371
CompressiveStrength	Age	0.337371
Water	Age	0.279269
Age	Water	0.279269
dtype: float64		
Tail		

BlastFurnaceSlag	Cement	-0.303301
Cement	BlastFurnaceSlag	-0.303301
FlyAsh	BlastFurnaceSlag	-0.312342
BlastFurnaceSlag	FlyAsh	-0.312342
Cement	FlyAsh	-0.385618
FlyAsh	Cement	-0.385618
FineAggregate	Water	-0.444888
Water	FineAggregate	-0.444888
Superplasticizer	Water	-0.646875
Water	Superplasticizer	-0.646875
dtype: float64		

Gambar 2.10 Urutan nilai korelasi antar atribut

Gambar 2.10 menampilkan nilai korelasi antar atribut dalam dataset. Terdapat nilai korelasi positif yang menunjukkan bahwa dua variabel memiliki hubungan positif jika nilainya mendekati 1. Terdapat pula nilai korelasi negatif jika nilai korelasinya semakin mendekati -1. Dengan mengurutkan nilai korelasi dari yang paling signifikan (baik yang positif maupun yang negatif), kita dapat lebih memusatkan perhatian pada korelasi yang kuat dengan variabel target, serta hubungan yang signifikan antara atribut lainnya.

6) Visualisasi *scatter plot* yang memvisualisasikan hubungan antara atribut "*Cement*," "*Age*," dan "*Water*" dengan atribut target "*Compressive Strength*"

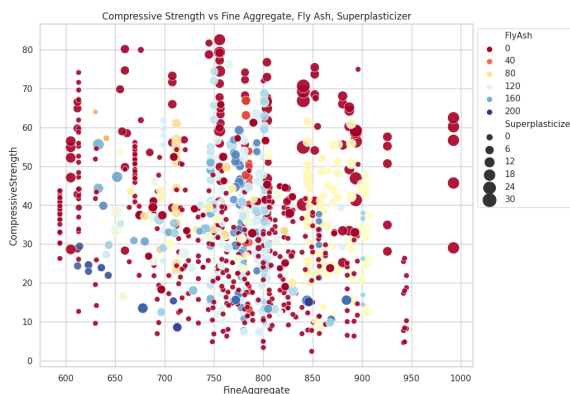


Gambar 2.10 Scatter plot yang memvisualisasikan hubungan antara atribut "*Cement*," "*Age*," dan "*Water*" dengan atribut target "*Compressive Strength*"

Dari *scatter plot* pada gambar 2.10, dapat diamati beberapa hubungan antara atribut-atribut berikut:

- Terdapat korelasi positif antara *Compressive Strength* dan *Cement*.
- Terdapat korelasi positif antara *Compressive Strength* dan *Age*.
- Dapat dilihat bahwa untuk nilai *Age* yang tinggi pada *Cement*, biasanya diperlukan jumlah *Water* yang lebih banyak, seperti yang ditunjukkan oleh data point berwarna hijau dengan ukuran yang lebih besar.
- Terdapat korelasi negatif antara *Compressive Strength* dan *Water*.
- Dapat diperhatikan bahwa *Compressive Strength* yang tinggi pada nilai *Age* yang rendah memerlukan jumlah *Cement* yang lebih tinggi.

7) Visualisasi *scatter plot* antara atribut "*Fine Aggregate*", "*Fly Ash*," dan "*Superplasticizer*" dengan atribut target "*Compressive Strength*"



Gambar 2.11 scatter plot hubungan antara atribut "*Fine Aggregate*," "*Fly Ash*," dan "*Superplasticizer*" dengan atribut target "*Compressive Strength*"

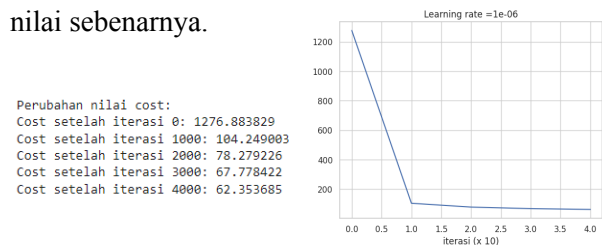
Berdasarkan gambar 2.11, dapat diamati keterkaitan antara komponen-komponen yang dapat digunakan untuk mengantisipasi campuran bahan yang optimal guna menciptakan beton yang memiliki kekuatan yang tinggi. Kemungkinan

besar, campuran beton yang ideal, dengan fokus pada peningkatan kekuatan, akan terdiri dari:

- Mengandung semen dalam jumlah besar.
- Rentang umur beton yang berlangsung lama, memerlukan banyak air namun hal ini pada akhirnya akan berpengaruh negatif pada kekuatan beton.
- Mengandung Superplasticizer dalam jumlah besar

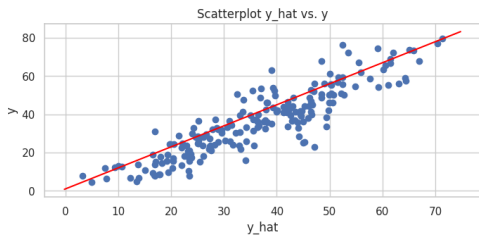
2.3. Pembangunan Model *Baseline*

Pada baseline model, training dilakukan sebanyak 5000 iterasi dengan menggunakan 3 hidden layer yang masing-masing memiliki 30 neurons, 20 neurons, dan 10 neurons serta tingkat pembelajaran sebesar 0.000001. Setelah pelatihan, prediksi dibuat untuk kedua set data, yaitu data tarin dan data test. Hasil prediksi ini digunakan untuk menghitung *Mean Squared Error* (MSE), yang mengukur seberapa dekat prediksi dengan data aktual. Ini memberikan gambaran bagus tentang seberapa baik model dapat mengantisipasi nilai sebenarnya.

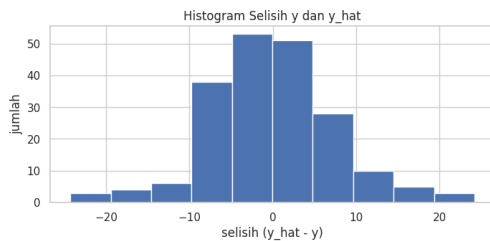


Gambar 2.12 Perubahan nilai cost per-iterasi pada baseline model

Pada gambar 2.12 dapat dilihat bahwa setiap iterasi dalam proses pelatihan model, nilai *cost* (biaya) cenderung menurun secara signifikan. Menurunnya biaya ini menunjukkan bahwa model semakin mendekati performa optimal saat melakukan prediksi terhadap data latih. Semakin rendah biaya pada iterasi terakhir, semakin baik model dalam melakukan prediksi dan semakin tepat model dalam memodelkan hubungan antara variabel yang diamati.



Gambar 2.13 Scatter Plot antara nilai prediksi (y_{hat}) dan nilai aktual (y)



Gambar 2.14 Histogram selisih nilai prediksi (y_{hat}) dan nilai aktual (y)

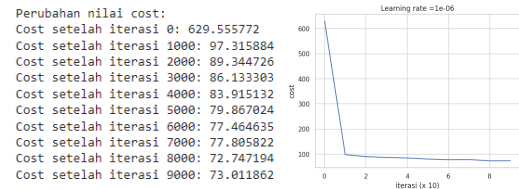
Dari pelatihan baseline model, didapatkan nilai MSE pada data latih adalah 233.84, sedangkan pada data uji adalah 59.08. Rendahnya nilai MSE data uji menunjukkan bahwa model *baseline* ini adalah model yang cukup akurat untuk memprediksi kuat tekan beton. Meskipun demikian, nilai MSE pengujian yang jauh lebih rendah dibanding nilai MSE pelatihan menunjukkan bahwa terdapat kejanggalan karena normalnya nilai MSE pengujian akan lebih tinggi dibandingkan nilai MSE pelatihan.

2.4. Eksplorasi Model

1) Model 1

Pada Model 1, dilakukan eksplorasi model dengan memperbesar jumlah iterasi (epoch) menjadi 10,000. Dengan demikian, model lebih banyak "belajar" dari data latih dalam proses pelatihannya. Adapun *hidden layer* yang digunakan adalah 3 *layer* yang masing-masing memiliki 30 *neurons*, 20 *neurons*, dan 10

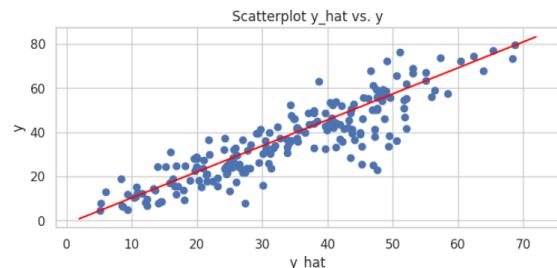
neurons serta tingkat pembelajaran sebesar 0.000001.



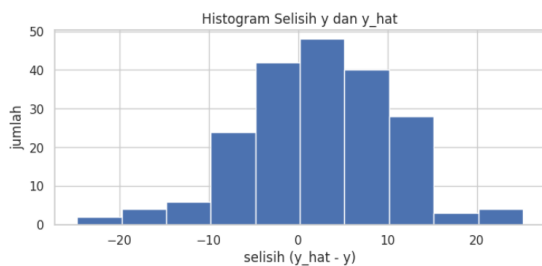
Gambar 2.15 Perubahan nilai cost per-iterasi pada model 1

Pada gambar 2.15, terlihat perubahan nilai cost dimana perbaikan model seiring dengan iterasi yang meningkat. Awalnya, cost yang tinggi pada iterasi pertama secara bertahap menurun ke level yang lebih rendah. Perubahan ini menunjukkan peningkatan dalam kemampuan model untuk memprediksi secara lebih akurat seiring dengan proses belajarnya. Meskipun terjadi sedikit fluktuasi di beberapa titik, secara keseluruhan, model mengalami perbaikan dalam kinerjanya seiring dengan iterasi yang lebih banyak.

Pada model 1, nilai MSE *Training* sebesar 281.31 dan MSE Test sebesar 71.86, yang menunjukkan model memiliki kinerja lebih baik pada data latih dibanding data uji. Idealnya, MSE test yang lebih rendah menunjukkan model yang lebih baik dalam memprediksi data baru. Meskipun demikian, nilai MSE model 1 lebih tinggi dibandingkan model *baseline*.



Gambar 2.16 Scatter Plot antara nilai prediksi (y_{hat}) dan nilai aktual (y) model 1

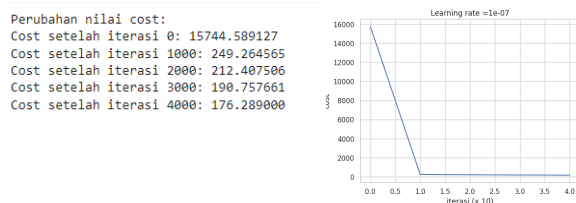


Gambar 2.17 Histogram antara nilai prediksi ($y_{\hat{}}$) dan nilai aktual (y) model 1

Dari pelatihan model 1, didapatkan nilai MSE pada data latih adalah 281.31, sedangkan pada data uji adalah 71.86 yang menunjukkan model memiliki kinerja lebih baik pada data latih dibanding data uji. Idealnya, MSE test yang lebih rendah menunjukkan model yang lebih baik dalam memprediksi data baru.

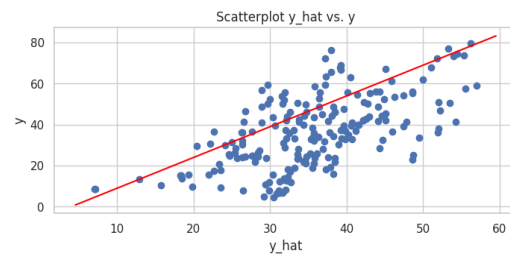
2) Model 2

Pada Model 2, dilakukan eksplorasi model dengan merubah learning rate yang diperkecil seper-10 menjadi 0.0000001 dan jumlah iterasi sebanyak 5000. Adapun hidden layer yang digunakan adalah 3 layer yang masing-masing memiliki 30 neurons, 20 neurons, dan 10 neurons.

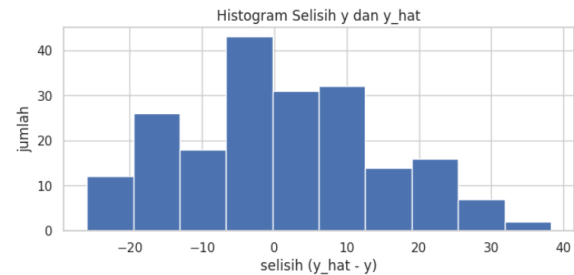


Gambar 2.18 Perubahan nilai cost per-iterasi pada model 2

Pada gambar 2.18, terlihat perubahan nilai cost menunjukkan model mengalami peningkatan performa dengan learning rate yang lebih kecil dengan jumlah iterasi yang lebih kecil dari Model 1 yaitu sebesar 5000. Hasil ini bisa menandakan potensi konvergensi yang lebih baik dengan kemungkinan peningkatan performa model.



Gambar 2.19 Scatter Plot antara nilai prediksi ($y_{\hat{}}$) dan nilai aktual (y) model 2



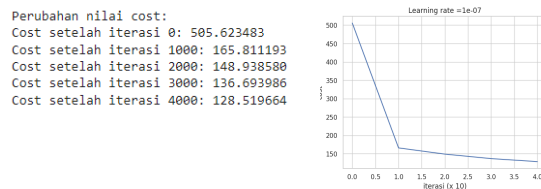
Gambar 2.20 Histogram antara nilai prediksi ($y_{\hat{}}$) dan nilai aktual (y) model 2

Pada model 2 melibatkan lapisan utama yang terdiri dari input maupun output serta lapisan tersembunyi lainnya, *hidden layer neurons* dan *epoch*. Pada model ini merupakan konfigurasi model yang berbeda, yang mencoba mengatur menjadi lebih banyak jumlah neuron dan lapisan tersembunyi lainnya. Hasil kinerjanya lebih baik jika dibandingkan dengan Model 1. Dengan MSE, model yang memiliki MSE lebih kecil pada data uji cenderung lebih baik karena mampu memberikan prediksi yang akurat. Pada model 2 ini memiliki MSE yang lebih rendah pada fase pelatihan maupun pengujian dibandingkan model baseline.

3) Model 3

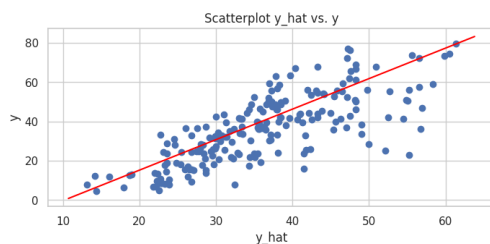
Model 3 mengalami eksplorasi yang signifikan dalam strukturnya. Berbeda dengan Model 1 dan 2 yang hanya menggunakan 3 *layer* dengan jumlah *neurons* 30, 20, dan 10, Model 3 mengadopsi 6 *layer* dengan perubahan jumlah neuron pada setiap *layer* menjadi 60, 50, 40, 30, 20, dan 10. Sementara itu, Model 2

menggunakan *learning rate* 0.0000001 dan berjalan selama 5000 iterasi.

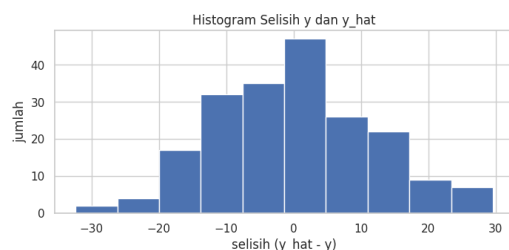


Gambar 2.21 Perubahan nilai cost per-iterasi pada model 3

Pada gambar 2.21 terlihat penurunan *cost* sepanjang iterasi. Perubahan ini menunjukkan peningkatan performa model terhadap prediksi, dan perubahan struktur model pada Model 3 memberikan hasil yang lebih baik daripada Model 1 dan 2. Meskipun menggunakan *learning rate* yang lebih kecil, Model 3 mampu secara efektif mengurangi *cost* lebih rendah pada iterasi 5000 dibandingkan dengan Model 2.



Gambar 2.22 Scatter Plot antara nilai prediksi (\hat{y}) dan nilai aktual (y) model 3



Gambar 2.23 Histogram antara nilai prediksi (\hat{y}) dan nilai aktual (y) model 3

Hasil evaluasi model 3 menunjukkan MSE pada data latih sebesar 487.31, sementara pada data uji sebesar 134.99. Ini menandakan bahwa model mungkin terlalu kompleks, menyebabkan *overfitting* pada data latih, namun juga memiliki performa yang baik pada data uji.

2.5. Evaluasi

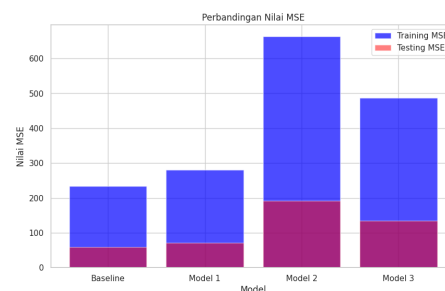
Berdasarkan hasil eksplorasi model di atas, didapatkan nilai sebagai berikut :

MSE Training Model Baseline : 233.84686592046324
 MSE Testing Model Baseline : 59.08636836820096

MSE Training Model 1 : 281.31809490038637
 MSE Testing Model 1 : 71.86349525774575

MSE Training Model 2 : 663.1222094483494
 MSE Testing Model 2 : 192.39342780987064

MSE Training Model 3 : 487.3122138425051
 MSE Testing Model 3 : 134.98638336682373



Gambar 2.24 Evaluasi model

Dari hasil evaluasi keempat model di atas, diperoleh perbandingan *Mean Squared Error* (MSE) dari empat model berbeda, yaitu model *baseline*, Model 1, Model 2, dan Model 3. Untuk setiap model tersebut, terdapat MSE untuk kedua fase, yaitu fase pelatihan (*training*) dan fase pengujian (*testing*). MSE adalah metrik yang digunakan untuk mengukur seberapa baik model regresi dalam menyesuaikan data, dimana semakin rendah MSE, maka semakin baik kinerja model.

3. KESIMPULAN

Berdasarkan evaluasi serta analisis hasil konfigurasi model, kita dapat lebih memahami mengapa model dengan konfigurasi yang berbeda memiliki performa yang berbeda juga. Model *baseline* adalah model dengan kinerja yang terbaik, ini dikarenakan model *baseline* memiliki keseimbangan antara nilai-nilai *hyperparameter*-nya. Sementara itu, model 1 yang memiliki epoch lebih

besar, mungkin mengalami *overfitting*. Selanjutnya, model 2 yang memiliki nilai learning rate yang lebih kecil mungkin mengindikasikan bahwa epoch-nya tidak seimbang. Terakhir, model 3 yang memiliki jumlah hidden layer dan jumlah neuron yang lebih banyak memerlukan pengelolaan yang lebih baik untuk menghindari *overfitting*. Pada keempat model tersebut terdapat persamaan berupa tidak seimbangnya nilai MSE pengujian dan nilai MSE pelatihan. Hal tersebut mungkin disebabkan oleh pembagian data yang kurang tepat antara data pengujian dan pelatihan. Dengan demikian, pembagian data yang tepat dan penyesuaian konfigurasi model untuk mendapatkan nilai *hyperparameter* yang seimbang adalah cara yang penting dilakukan untuk meningkatkan kinerja model regresi, baik dalam kasus ini maupun kasus-kasus lainnya.

4. DAFTAR PUSTAKA

- [1] Yeh I-C (1998). Pemodelan kekuatan beton kinerja tinggi menggunakan jaringan saraf tiruan. *Cement and Concrete Research*, 28(12), 1797-1806.
- [2] Haykin S (1999). *Jaringan Saraf: Dasar-Dasar Komprehensif*. Upper Saddle River, N.J, AS: Prentice-Hall.
- [3] R. Olawoyin, "Application of backpropagation artificial neural network prediction model for the PAH bioremediation of polluted soil," *Chemosphere*, 2016.