

# Klasifikasi Kekuatan Struktur Beton Menggunakan Convolutional Neural Networks

Johan Hartanto<sup>1</sup>, Lina<sup>2</sup>

<sup>1,2)</sup> Teknik Informatika, FTI, Universitas Tarumanagara  
Jl. Letjen S Parman no 1, Jakarta 11440 Indonesia  
email : <sup>1</sup>Johan.535180119@stu.untar.ac.id <sup>2</sup>lina@fti.untar.ac.id

## ABSTRACT

*Concrete is one of the most important elements in building a building construction. Concrete is widely used because it has advantages compared to other construction materials. In addition, the development of concrete construction has increased rapidly compared to other constructions, especially in the way of making concrete to the technology and use of materials used. In its development, materials will increase so that experiments in the laboratory make the costs swell. Therefore, a research is proposed which is intended to help researchers as well as to provide a comparison of the use of the model used.*

*The method used to classify will use the CNN model by producing output that will display the class categories on the variables that have been inputted. The test results on training data resulted in an accuracy of 86.04% and testing on test or validation data was 82.14% on the Adam optimizer and 83.25% on training data and 80.35% on test or validation data on RMSprop. After determining the model to be used, it is continued with the use of K-fold validation.*

## Key words

*Concrete, Convolutional Neural Network, Classification of Compressive Strength of Concrete Structures, Confusion Matrix, K-Fold*

## 1. Pendahuluan

Beton merupakan salah satu unsur yang sangat penting dalam membangun sebuah konstruksi bangunan. Beton banyak digunakan karena memiliki keunggulan dibandingkan dengan bahan konstruksi lainnya. Selain itu, perkembangan konstruksi beton meningkat pesat dibandingkan dengan konstruksi lainnya terutama pada cara pembuatan beton hingga teknologi dan penggunaan bahan yang digunakan. Teknologi akan digunakan untuk mencoba berbagai metode dalam perencanaan rancangan campuran beton. Menurut Nikoo pada tahun 2015, menyatakan bahwa model campuran beton yang dibuat manual dengan menggunakan banyak kombinasi bahan dan pengujian secara destruktif membutuhkan biaya yang besar dan tidak akurat. Model komputasi diperlukan untuk memperoleh desain campuran beton yang lebih akurat sesuai dengan perilaku beton yang dikehendaki[1].

Umumnya beton terbuat dari berbagai campuran bahan konstruksi yang meliputi air, semen, agregat kasar dan agregat halus. Untuk menghasilkan beton dengan kualitas tinggi penggunaan bahan tidak dapat selalu dicapai dengan hanya menggunakan material konvensional.

Untuk mengetahui seberapa kuat beton yang dihasilkan, akan di uji melalui laboratorium dengan menggunakan alat menguji tekanan yang bernama *Compression Testing Machine*. Pengujian kuat tekanan beton akan menghasilkan *output sample* yang berjumlah puluhan hingga ribuan untuk mendapatkan data kuat tekan beton yang kredibel dan bervariasi sesuai dengan presentase jumlah bobot campuran bahan beton pada rancangan campuran beton. Lebih detail mengenai proses uji kuat tekan beton terdaftar pada sertifikat SNI 1974:2011 yang mengenai standar uji kuat tekan beton dengan benda uji silinder yang menjadi standar nasional Indonesia bidang pengujian[2].

Dengan penggunaan teknologi ini, akan sangat menarik jika menggunakan arsitektur CNN (*Convolutional Neural Networks*) yang merupakan turunan dari Neural Network biasa. Penggunaan CNN dalam proses klasifikasi kuat tekan beton diharapkan dapat mampu membantu pengguna dalam menentukan bahan baku beton yang pas guna membentuk beton dengan kualitas yang diharapkan. Penggunaan *Softmax Classifier* untuk permasalahan klasifikasi, dan umumnya digunakan pada *output layer*. Selanjutnya akurasi klasifikasi untuk mengukur kinerja model dengan rumus akurasi sama dengan jumlah prediksi yang benar dibagi dengan total jumlah sample input. Perancangan sistem ini diharapkan mampu untuk melakukan peng-klasifikasian nilai kuat tekan beton yang terdiri dari tekan kuat, tekan sedang, dan tekan lemah.

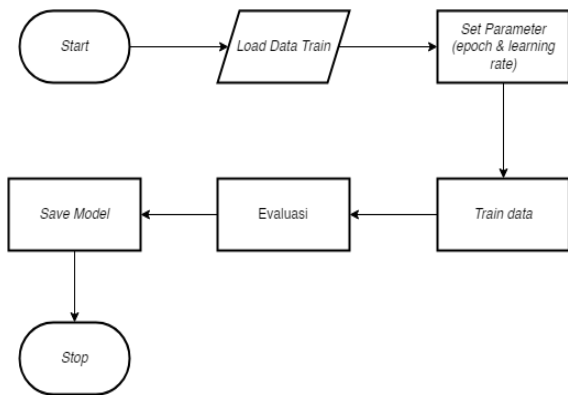
## 2. Dasar Teori

Sistem yang dirancang diharapkan dapat mampu melakukan klasifikasi pada kekuatan struktur beton dengan menggunakan *Convolutional Neural Networks*. Sistem ini diharapkan akan memperlihatkan hasil yang berbentuk angka numerik beserta kelas yang sudah

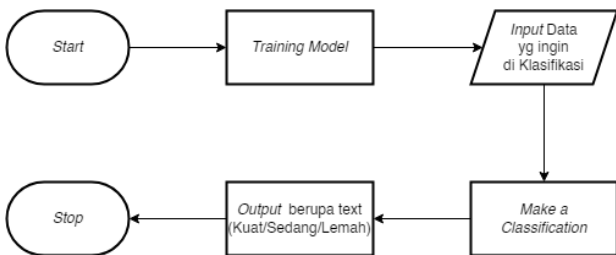
ditentukan sebelumnya sebagai indikasi bahwa tiap-tiap data menghasilkan angka yang tepat sasaran yaitu sesuai dengan kelas yang diharapkan. Sistem yang dirancang akan memiliki tampilan *User Interface* untuk memasukkan data testing sehingga *user* dapat menggunakan aplikasi tersebut dan data training beserta model CNN sudah dilakukan sebelumnya. Ilustrasi *flowchart* CNN terdapat pada **Gambar 1**. Ilustrasi *flowchart* training pada **Gambar 2**, dan ilustrasi *flowchart* testing pada **Gambar 3**.



Gambar 1. Flowchart CNN.



Gambar 2. Flowchart Training.



Gambar 3. Flowchart Testing

2.1 Pengumpulan Dataset

Untuk pengujian model dilakukan dengan menggunakan dataset yang telah di peroleh dari sumber. Dataset akan dibagi menjadi 2 secara otomatis dengan menggunakan library sklearn yaitu train test split. Dengan mengatur nilai dari test size, dataset akan dibagi menjadi 2 yaitu dataset train dan dataset testing/validation sesuai dengan nilai yang ditentukan. Dataset train akan digunakan untuk melatih model dan dataset test/validation digunakan sebagai validasi atau testing model yang telah dibuat dengan menggunakan dataset real world atau dataset yang tidak ada di dalam model. Jumlah perbandingan komposisi per kelas ditunjukkan pada Tabel 1.

Kelas	Jumlah
1 – Kuat	434
2 – Sedang	484
3 – Lemah	202
Total	1120

Tabel 1. Komposisi Dataset per kelas

Sebanyak 1120 sampel beton yang akan dievaluasi dari laboratorium penelitian universitas pembangunan jaya digunakan untuk menguji model prediksi. Semua tes dilakukan dengan silinder yang dibuat dengan menggunakan prosedur SNI 1974:2011. Tabel 2 menunjukkan set data eksperimen yang berjumlah 8 buah variable yang akan digunakan dalam penelitian ini.

Variabel	Nilai
X1 = Cement (kg/m <sup>3</sup> )	168
X2 = Slag (kg/m <sup>3</sup> )	42.1
X3 = Ash (kg/m <sup>3</sup> )	163.8
X4 = Water (kg/m <sup>3</sup> )	121.8
X5 = Super Plastic (kg/m <sup>3</sup> )	5.7
X6 = Coarse Aggregate (kg/m <sup>3</sup> )	1058.7
X7 = Fine Aggregate (kg/m <sup>3</sup> )	780.1
X8 = Age (hari)	14

Tabel 2. Variabel Data Beton

2.2 Convolutional Neural Networks

Dalam mencapai hasil akurasi model yang maksimal, dilakukan percobaan dengan perbandingan arsitektur cnn dengan menggunakan 1 hidden layer dan aktivasi tanh dengan 1 atau 2 hidden layer dan menggunakan aktivasi reLu. Hasilnya penggunaan aktivasi reLu dan penambahan hidden layer berbuah manis dengan adanya peningkatan signifikan. Hasil tersebut dapat dilihat pada Tabel 3.

Epoch = 1000				
	1 Hidden Layer (tanh)	1 Hidden Layer (reLu)	2 Hidden Layer (tanh)	2 Hidden Layer (reLu)
Train Accuracy	73.01%	74.69%	79.06%	81.95%
Test/Val Accuracy	75%	76.06%	74.10%	82.14%

Tabel 3. Hasil perbandingan hidden layer dan activation

Dalam perancangan model dengan tingkat akurasi yang maksimal, dibutuhkan penggunaan optimizer dengan tujuan untuk membantu meminimalkan fungsi kesalahan (loss or error function). Pengaturan learning rate dalam optimizer juga sangat membantu dalam menentukan hasil yang di dapat. Variasi learning rate akan berada pada nilai 0.1, 0.01, dan 0.001 yang merupakan nilai default pada optimizer. Pemilihan optimizer dan penentuan learning rate dengan epoch sebesar 500 dapat dilihat pada Tabel 4.

Optimizer	Epoch	Learning Rate		
		0.001	0.01	0.1
Adam	100	53.66%	57.88%	40.09%
	250	71.43%	39.64%	43.57%
	500	77.37%	42.31%	40.63%
RMSprop	100	55.98%	42.85%	42.05%
	250	67.22%	43.84%	41.96%
	500	74.28%	49.54%	40.98%
SGD	100	42.94%	43.22%	43.22%
	250	43.21%	43.39%	43.22%
	500	43.30%	43.30%	43.21%

Tabel 4 Optimizer dan Learning Rate berdasarkan baseline

Dapat dilihat pada tabel 3 bahwa penggunaan optimizer Adam dan RMSprop pada epoch 500 menghasilkan tingkat akurasi (berdasarkan baseline) sebesar 75%. Penggunaan epoch sebesar 500 menghasilkan nilai yang cukup, maka dari itu peningkatan epoch pada percobaan variasi hyperparameter berikutnya akan di tingkatkan menjadi 1000 dan 1200 epoch guna mendapatkan hasil yang lebih baik lagi dibandingkan penggunaan dengan epoch 500.

Percobaan model dengan berbagai *hyperparameter* menghasilkan nilai tertinggi dengan penggunaan *optimizer* Adam dan RMSprop, menggunakan *learning rate* sebesar 0.001. Penggunaan *learning rate* ini ditujukan untuk menghitung nilai koreksi bobot pada saat dilakukan proses *training*. Dimana, jika nilai lr semakin besar maka proses *train* akan berjalan semakin cepat dan sebaliknya. Namun penggunaan lr yang besar dapat menyebabkan kerugian seperti tingkat ketelitian pada jaringan yang akan semakin berkurang.

Hasilnya pada percobaan tersebut penggunaan epoch sebesar 1200 lebih baik dibandingkan dengan menggunakan 1000 saja. Namun, dapat terlihat bahwa penggunaan tersebut tidak menghasilkan peningkatan yang signifikan. Hasil dapat dilihat pada Tabel 5.

Learning Rate = 0.001				
Epoch	1000		1200	
Adam optimizer				
Test Size	0.1	0.2	0.1	0.2
Train Accuracy	80.95%	86.04%	81.94%	82.70%

Test/Val Accuracy	82.14%	79.46%	80.35%	76.78%
RMSprop optimizer				
Test Size	0.1	0.2	0.1	0.2
Train Accuracy	64.98%	63.39%	82.83%	83.25%
Test/Val Accuracy	67.85%	58.48%	80.35%	78.57%

Tabel 5 Hasil Adam dan RMSprop

### 2.3 Confusion Matrix

Untuk menentukan hyperparameter yang akan digunakan sebagai model sesungguhnya, akan dilihat dari parameter pendukung seperti nilai *accuracy*, *precision*, *recall* dan *f1 score* pada *confusion matrix*. Akan ditampilkan keseluruhan percobaan yang telah dihasilkan pada tabel-tabel diatas dengan hasil grafik *train* dan *validation accuracy* berserta dengan tampilan dari *confusion matrix*. *Confusion matrix* merupakan metode yang dapat mengukur performa dalam melakukan klasifikasi pada *machine learning*.

Namun dalam penggunaan *multi-class classification* tidak ada yang namanya kelas *positive* atau *negative*. Tetapi itu tidak membuat penggunaan *confusion matrix* ini terhalang, karena penggunaan untuk *multi-class classification* mudah. Untuk menghasilkan nilai acuan dalam kinerja dari penggunaan *confusion matrix* adalah mendapatkan nilai TP, TN, FP, FN setiap kelas dan menggunakannya untuk memperoleh nilai yang dihasilkan dari model yang dievaluasi, nilai tersebut akan menghasilkan variabel akurasi, presisi, recall, dan *f1 score*. Berikut merupakan deskripsi singkat mengenai ukuran kinerja dalam *confusion matrix*, yaitu:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Accuracy, memberikan akurasi model secara keseluruhan yang memiliki fraksi dari total sampel yang diklasifikasikan dengan benar oleh pengklasifikasi.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Precision, tolak ukur terhadap seberapa baik model dalam memprediksi data benar positif (TP) di antara keseluruhan data yang terprediksi positif

$$Recall = \frac{TP}{RP + FN} \tag{3}$$

Recall, tolak ukur terhadap seberapa akurat model dalam memprediksi data benar positif diantara keseluruhan data yang benar positif.

$$f1\ score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

$$= \frac{2TP}{2TP + FP + FN}$$

F1 Score, tolak ukur yang menggabungkan precision dan recall menjadi satu ukuran nilai. Secara matematis, ini merupakan rata-rata dari precision dan recall.

### 2.3 K-Fold

Setelah semua ditentukan, selanjutnya akan dilakukan tahap penggunaan *k-fold validation*. Sebenarnya ini merupakan salah satu cara dalam melakukan evaluasi model yang dibuat, karena sebelumnya sudah menggunakan evaluasi lain seperti *confusion matrix*. Namun, penggunaan ini tetap dilakukan karena dapat membantu dalam memperlihatkan hasil dari pemilihan *hyperparameter* sebelumnya. Akan menggunakan Adam dan RMSprop dengan lr sebesar 0.001 dengan hasil Tabel 6.

Optimizer	Iterasi Fold Ke-	Hasil Loss	Hasil Akurasi
Adam (0.001)	1	10.96%	83.66%
	2	0.5%	68.66%
	3	34.25%	67.23%
	4	1%	71.96%
	<b>Rata-rata</b>	<b>11.70%</b>	<b>72.87%</b>
RMSprop (0.001)	1	29.09%	85.80%
	2	11.72%	82.86%
	3	15.25%	85.80%
	4	8.9%	85.36%
	<b>Rata-rata</b>	<b>16.26%</b>	<b>84.95%</b>

Tabel 6 Hasil k-fold

### 3. Hasil Percobaan

Berdasarkan pengujian yang telah dilakukan terhadap model dan program akan dituangkan pada bagian ini. Proses pelatihan model akan menggunakan *train test split* dengan perbandingan data latih sebesar 80% dan data uji sebesar 20%. Pembuatan model tersebut dimaksudkan agar keseluruhan kelas pada dataset yang dimasukkan ke dalam model akan lebih banyak dikenali. Adanya sedikit ketidak seimbangan data juga mempengaruhi jumlah akurasi yang didapat. Dihasilkan akurasi yang didapat mencapai 85% dan saat tahap *test*

atau evaluasi juga menghasilkan angka presentase yang cukup.

### 4. Kesimpulan

Setelah melewati proses penelitian dan percobaan dalam pembuatan program pengklasifikasian kekuatan tekan beton ini dapat disimpulkan bahwa :

1. Penelitian ini menyajikan studi perbandingan antara penggunaan berbagai hyperparameter dan sebagai tolak ukur dalam penggunaan arsitektur ini pada dataset beton dengan 3 buah jenis tingkatan kekuatan beton yaitu kuat, sedang, dan lemah dengan jumlah 1120 data sampel. Terbagi menjadi 3 yaitu, kelas 1 sebanyak 434 baris, kelas 2 sebanyak 484 baris, dan kelas 3 sebanyak 202 baris. Dari hasil tersebut akan menggunakan optimizer RMSprop yang merupakan parameter optimal dengan tingkat akurasi sebesar 84.95% pada K-Fold, 83.25% akurasi train, dan 78.57% akurasi test/validation.
2. Terdapat beberapa indikator akurasi yang digunakan dalam mengevaluasi performa dari metode yang digunakan, yaitu MAE, MSE, RMSE, R, *Confusion Matrix*, dan *K-Fold Validation*. Percobaan dilakukan dengan menggunakan bantuan *software* Google Colab.
3. Dari hasil eksperimen yang didapat bahwa penggunaan *optimizer* Adam dan RMSprop tidak jauh berbeda dibandingkan dengan yang lain. RMSprop lebih unggul sedikit dibandingkan dengan Adam yang telah dilakukan percobaan menggunakan 1000 dan 1200 *epoch* serta didukung variasi *hyperparameter*.
4. Hasil blackbox testing pada setiap modul yang dirancang sudah berjalan dengan semestinya. Semua fungsi serta fitur yang di implementasikan sudah berjalan dengan baik.
5. Modul klasifikasi kuat tekan beton menggunakan algoritma CNN dengan arsitektur utama yaitu LeNet dan beberapa konfigurasi demi meningkatkan performa model telah dilakukan dan didapatkan akurasi yang cukup baik untuk melakukan klasifikasi.
6. Penelitian ini berhasil membuktikan setidaknya bahwa penggunaan metode CNN dapat melakukan klasifikasi pada kuat tekan beton tanpa harus melakukan percobaan dan eksperimen secara langsung di laboratorium.

### REFERENSI

- [1] BSN; Badan Standarisasi Nasional, SNI 1974:2011 tentang Cara uji kuat tekan beton dengan benda uji silinder, <https://binamarga.pu.go.id/index.php/nspk/detail/sni-19742011-tentang-cara-uji-kuat-tekan-beton-dengan-benda-uji-silinder>, 29 Maret 2022.
- [2] Nikoo, Mehdi; Moghadam Farshid Torabian; dan Sadowski Aukasz, "Prediction of Concrete

Compressive Strength by Evolutionary Artificial Neural Networks”, *Advances in Materials Science and Engineering*, Vol. 2015, 2015, <http://dx.doi.org/10.1155/2015/849126>.

**Johan Hartanto**, saat ini sebagai mahasiswa Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Tarumanagara.

**Lina S.T., M.Kom., Ph.D.**, memperoleh gelar Sarjana dari Universitas Tarumanagara, Indonesia tahun 2001 dan gelar Magister dari Universitas Indonesia, Indonesia tahun 2004. Kemudian tahun 2009 memperoleh gelar Ph.D. dari Nagoya University, Jepang. Saat ini sebagai Dosen Tetap Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Tarumanagara.