



OPTIMASI KUAT TEKAN BETON DENGAN JARINGAN SYARAF TIRUAN METODE *MULTI LAYER PERCEPTRON*

(OPTIMIZATION OF CONCRETE COMPRESSIVE STRENGTH WITH ARTIFICIAL NEURAL NETWORKS USING MULTI LAYER PERCEPTRON)

Lutfan Anas Zahir¹, Suliana Mafiroh²

¹Program Studi Teknik Sipil, Fakultas Teknik, Universitas Tulungagung
Alamat korespondensi:

E-mail: lutfananas@gmail.com

²Program Studi Teknik Sipil, Fakultas Teknik, Universitas Tulungagung
Alamat korespondensi:

E-mail: sulianamafiroh@gmail.com

Abstract

The optimization of concrete compressive strength is crucial for the development of durable and high-performance concrete structures. This study investigates the application of Artificial Neural Networks (ANN) using Multi-Layer Perceptron (MLP) regression techniques to optimize the mix design of concrete. By leveraging MLP, a type of ANN known for its ability to model complex non-linear relationships, we aim to predict and enhance the compressive strength of concrete based on various mix proportions. The study utilizes a dataset comprising different mix designs, including cement, water, aggregates, and admixtures. The MLP model is trained and validated using this dataset, demonstrating its capability to accurately predict concrete compressive strength. Results indicate that the MLP-based ANN model outperforms traditional mix design methods, providing more precise and reliable predictions. This approach not only enhances the understanding of the relationships between mix components and compressive strength but also offers a robust tool for engineers to optimize concrete mixes effectively. The findings highlight the potential of integrating advanced computational techniques in civil engineering to achieve superior material performance..

Keywords: *Optimization, Artificial Neural Networks, Multi-Layer Perceptron, Concrete Mix Design, Construction Materials*

Abstrak

Optimasi kuat tekan beton sangat penting untuk pengembangan struktur beton yang tahan lama dan berkinerja tinggi. Studi ini meneliti penerapan Jaringan Syaraf Tiruan (JST) menggunakan teknik regresi Multi-Layer Perceptron (MLP) untuk mengoptimalkan desain campuran beton. Dengan memanfaatkan MLP, jenis JST yang dikenal karena kemampuannya untuk memodelkan hubungan non-linear yang kompleks, kami bertujuan untuk memprediksi dan meningkatkan kuat tekan beton berdasarkan berbagai proporsi campuran. Studi ini menggunakan dataset yang terdiri dari berbagai desain campuran, termasuk semen, air, agregat, dan bahan tambahan. Model MLP dilatih dan divalidasi menggunakan dataset ini, menunjukkan kemampuannya untuk memprediksi kuat tekan beton dengan akurat. Hasil penelitian menunjukkan bahwa model JST berbasis MLP mengungguli metode desain campuran tradisional, memberikan prediksi yang lebih tepat dan andal. Pendekatan ini tidak hanya meningkatkan pemahaman tentang hubungan antara komponen campuran dan kuat tekan, tetapi juga menawarkan alat yang kuat bagi para insinyur untuk mengoptimalkan campuran beton secara efektif. Temuan ini menyoroti potensi integrasi teknik komputasi canggih dalam teknik sipil untuk mencapai kinerja material yang unggul

Kata kunci: *Optimasi, Jaringan Syaraf Tiruan, Multi-Layer Perceptron, Desain Campuran Beton, Material Konstruksi*

PENDAHULUAN

Desain campuran beton merupakan proses kompleks yang terdiri dari berbagai tahap, di mana dilakukan upaya untuk mencari kombinasi bahan yang ideal guna menghasilkan beton dengan performa terbaik. Campuran dan kekuatan beton merupakan aspek penting dalam perancangan struktur beton bertulang (Hossein et al, 2022). Desain campuran beton yang efektif dapat memastikan beton mencapai kekuatan dan daya tahan yang dibutuhkan, yang sangat krusial dalam konstruksi bangunan beton bertulang (Pushkar, 2019). Proses ini mencakup analisis mendalam terhadap berbagai faktor seperti jenis dan proporsi agregat, rasio campuran air dan semen, aditif, dan

lainnya, dengan tujuan untuk memperoleh kualitas dan kekuatan beton yang terbaik. Beberapa studi independen telah menunjukkan bahwa kekuatan beton tidak hanya dipengaruhi oleh rasio air-campuran (w/c ratio), tetapi juga oleh kandungan bahan lainnya (Mater, 1994). Langkah awal dalam desain campuran beton melibatkan pemahaman yang mendalam tentang karakteristik material yang akan digunakan, serta evaluasi terhadap kebutuhan teknis proyek. Selanjutnya, parameter seperti kekuatan tekan, kerja (workability), ketahanan terhadap erosi, dan lain-lain, harus dipertimbangkan secara teliti untuk memastikan desain campuran beton memenuhi standar yang ditetapkan. Tujuan utama dari desain campuran beton adalah untuk menentukan komposisi kuantitatif dan proporsi yang tepat dari bahan campuran beton. Dalam proses pembuatan beton, campuran yang direncanakan harus dapat mencapai mutu beton yang optimal. Salah satu aspek mutu beton yang paling penting adalah kekuatan tekan. Desain campuran beton memegang peranan krusial dalam mencapai kualitas beton yang diinginkan, baik dari segi kekuatan maupun daya tahannya (Argiz et al, 2018).

Mostafaei et al. (2023) membahas dampak lingkungan dari desain campuran beton dan pentingnya mengoptimalkan desain campuran untuk mengurangi efek negatif, serta menggarisbawahi keterbatasan metode manual tradisional dalam desain campuran beton. Metode desain campuran beton konvensional masih memiliki beberapa kelemahan. Pada penelitian ini mengusulkan pendekatan baru untuk merancang campuran beton dengan menggunakan persamaan matematika yang dikembangkan melalui algoritma machine learning. Pendekatan machine learning ini dapat berfungsi sebagai alat untuk memperkirakan mutu beton secara kasar. Selama beberapa dekade terakhir, telah ada kebutuhan yang mendesak akan alat yang efektif untuk memprediksi kinerja beton, khususnya kekuatan tekanannya. Berbagai alat statistik telah diterapkan, termasuk persamaan regresi linear, multilinear, dan nonlinear. Selain itu, perkembangan teknologi machine learning baru-baru ini telah menarik perhatian signifikan, dengan prediksi masa depan teknologi ini semakin menjanjikan. Machine learning, sebagai cabang kecerdasan buatan, memungkinkan komputer untuk belajar dari data dan pengalaman, serta menjalankan tugas-tugas kompleks tanpa perlu diprogram secara eksplisit.

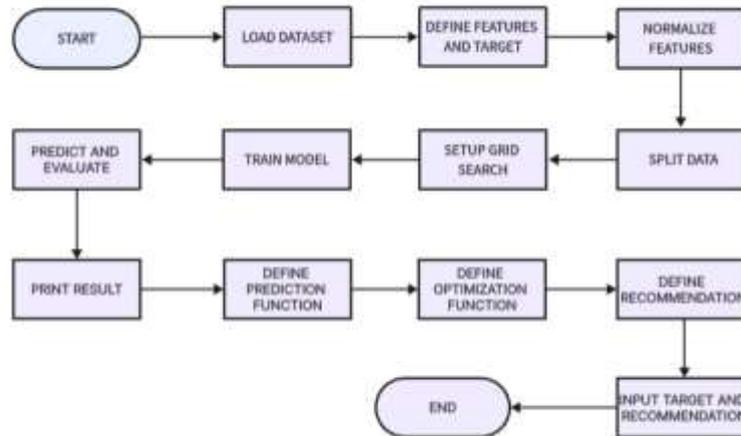
Beton adalah bahan konstruksi yang sangat penting dan banyak digunakan dalam industri konstruksi. Dalam berbagai aplikasi, mulai dari infrastruktur jalan raya, jembatan, hingga gedung pencakar langit, beton memainkan peran kunci dalam memastikan kekuatan dan stabilitas struktur. Salah satu parameter yang paling kritis dalam menentukan kualitas beton adalah kuat tekan, yaitu kemampuan beton untuk menahan beban kompresi. Kuat tekan beton yang optimal tidak hanya menjamin kekuatan struktur tetapi juga memperpanjang umur pakai dan mengurangi kebutuhan perawatan serta biaya pemeliharaan. Oleh karena itu, desain campuran beton yang tepat sangat penting untuk mencapai kuat tekan yang diinginkan.

Kamath et al. (2022) menemukan bahwa *Decision Regression Tree*, *Random Forest Regression*, *Lasso Regression*, *Ridge Regression*, dan *Multiple Linear Regression* menunjukkan performa terbaik dalam studi perbandingan lima model regresi *Machine Learning* (ML). Data set data historis dianalisis menggunakan lima model regresi ML yang berbeda, serta dua dataset dari laboratorium terkini. Kinerja model dievaluasi menggunakan koefisien determinasi (R^2), mean absolute error, root mean squared error, dan mean absolute percentage error. Model-model ini kemudian dibandingkan secara statistik dengan penelitian sebelumnya tentang topik tersebut. Selain itu, analisis sensitivitas dan pengujian parametrik digunakan untuk menilai efektivitas model prediktif dan memberikan wawasan mengenai dampak variabel-variabel tersebut. Kehadiran teknologi ini telah membawa perubahan signifikan dalam berbagai bidang, termasuk dalam desain campuran beton. Dalam konteks tersebut, machine learning dapat digunakan untuk menganalisis data besar terkait performa beton, sifat bahan, dan kondisi lingkungan, untuk memberikan rekomendasi komposisi campuran yang optimal. Hal ini menawarkan keunggulan dalam hal efisiensi dan akurasi desain, serta dapat menghemat waktu dan sumber daya.

METODE PENELITIAN

Datasets

Dataset yang digunakan dalam penelitian ini diperoleh dari UC Irvine Machine Learning Repository, yang menyediakan data yang diperlukan. Estimasi kekuatan tekan beton melibatkan lima variabel, yaitu semen, pasir, kerikil, air, kuat tekan. Korelasi antara distribusi variabel dan parameter yang digunakan untuk menghitung kekuatan tekan ditampilkan dalam Gambar berikut:



Gambar 1. Study Model

Flowchart program di atas mengilustrasikan alur dari pemrosesan data hingga memberikan rekomendasi komposisi beton. Proses dimulai dengan memuat data beton dari sumber yang tersedia, kemudian mendefinisikan fitur dan target yang akan digunakan dalam model. Langkah-langkah awal ini mencakup normalisasi fitur untuk memastikan bahwa data dalam skala yang sama, serta membagi data menjadi set latih dan set uji untuk melatih dan menguji model secara terpisah. Proses ini ditandai dengan kotak persegi panjang yang menggambarkan tahapan-tahapan yang harus dilakukan secara berurutan.

Pengaturan grid di search substitusikan untuk menentukan parameter terbaik bagi model MLPRegressor. Proses ini melibatkan pencarian parameter yang optimal melalui GridSearchCV, yang merupakan tahap penting untuk mengidentifikasi konfigurasi model yang paling efektif. Dalam hal ini, proses ini akan mencakup beberapa iterasi untuk menemukan parameter terbaik, yang digambarkan sebagai langkah pilihan dalam flowchart karena memerlukan keputusan yang mempengaruhi langkah selanjutnya, yaitu pelatihan model. Setelah model dilatih dengan parameter yang dioptimalkan, proses dilanjutkan dengan prediksi dan evaluasi hasil model. Model yang telah dilatih digunakan untuk memprediksi kekuatan tekan beton pada data uji, dan hasilnya dievaluasi menggunakan Mean Squared Error. Fungsi prediksi dan optimasi digunakan untuk merekomendasikan komposisi beton yang sesuai dengan target kekuatan tekan yang diinginkan. Tahapan ini diakhiri dengan memberikan rekomendasi komposisi kepada pengguna berdasarkan hasil optimasi. Flowchart berakhir dengan langkah "End," menandakan bahwa semua proses telah selesai.

Data Preprocessing

Pre-data processing adalah langkah penting untuk memastikan keakuratan, keandalan, dan integritas data yang akan digunakan dalam penelitian tentang penerapan Machine Learning (ML) dalam perencanaan mix design beton mutu tinggi. Proses ini mencakup langkah-langkah seperti pengumpulan data yang teliti, identifikasi potensi outliers, serta penanganan nilai yang hilang atau tidak konsisten. Selain itu, tahap pre-data processing juga melibatkan normalisasi atau standarisasi data, yang memastikan bahwa variabel-variabel dengan skala berbeda dapat diproses secara konsisten. Langkah ini membantu mengurangi potensi distorsi hasil yang dapat terjadi akibat dominasi variabel dengan skala besar, yang bisa mempengaruhi kinerja algoritma ML. Mengatur

data dengan standar yang sesuai adalah strategi penting untuk memaksimalkan efektivitas model ML yang akan dikembangkan. Dengan demikian, pre-data processing tidak hanya berfungsi sebagai persiapan data, tetapi juga sebagai strategi cerdas untuk mengoptimalkan validitas dan efektivitas model ML yang diharapkan, dengan menghasilkan hasil yang akurat dan berguna dalam perancangan mix beton mutu tinggi.

Data Preparation (Exploratory Data Analysis / EDA)

Persiapan data, atau yang lebih dikenal dengan istilah Exploratory Data Analysis (EDA), merupakan tahap krusial dalam penelitian mengenai penerapan Machine Learning (ML) untuk merancang mix design beton mutu tinggi. Proses EDA melibatkan analisis menyeluruh terhadap dataset yang telah dikumpulkan, mencakup identifikasi dan pemahaman pola data, distribusi variabel, serta korelasi kompleks antar variabel. Penulis melakukan eksplorasi mendalam mengenai hubungan antara parameter material beton, umur, dan kuat tekan beton, sehingga dapat memetakan interaksi yang rumit di antara variabel tersebut. Hasil analisis ini memberikan wawasan yang signifikan mengenai karakteristik penting yang menjadi dasar dalam merancang mix beton mutu tinggi. Selain itu, EDA memungkinkan penulis untuk mendeteksi anomali atau outlier yang bisa mempengaruhi hasil analisis ML, memastikan keakuratan dan keandalan model yang dikembangkan. Mengidentifikasi masalah tersebut, EDA berfungsi sebagai tahap penting dalam menjamin kualitas optimal dari dataset yang digunakan untuk membangun model ML. Akibatnya, proses EDA berperan penting dalam menghasilkan rencana mix design beton mutu tinggi yang akurat dan dapat diandalkan.

Apply the Algorithm

Tahapan ini berfokus pada penerapan berbagai algoritma machine learning untuk membangun model yang dapat menentukan rencana mix design beton mutu tinggi yang optimal. Langkah ini melibatkan implementasi sejumlah model algoritma machine learning yang relevan dengan konteks penelitian. Pemilihan algoritma yang tepat, disesuaikan dengan tujuan penelitian, menjadi faktor penting dalam memberikan kontribusi signifikan terhadap penelitian ini. Pada penelitian ini mempertimbangkan berbagai faktor seperti komposisi bahan yang mempengaruhi kualitas beton mutu tinggi. Penggabungan dan menganalisis informasi dari berbagai model, diharapkan dapat diperoleh wawasan mendalam mengenai cara mengoptimalkan kualitas beton mutu tinggi, serta meningkatkan efisiensi penggunaan sumber daya dan biaya produksi. Pendekatan ini mendukung industri konstruksi untuk mencapai tujuan berkelanjutan dan optimal, serta menghasilkan beton berkualitas tinggi untuk proyek-proyek infrastruktur dan bangunan yang baik.

Performance Measurement

Performance measurement dari algoritma machine learning dalam konteks perancangan mix design beton mutu tinggi melibatkan evaluasi sejauh mana model yang dikembangkan mampu memprediksi kekuatan tekan beton secara akurat. Beberapa metrik yang digunakan untuk mengukur kinerja model sebagai berikut:

1. Mean Squared Error (MSE)

MSE mengukur rata-rata kuadrat selisih antara nilai yang diprediksi dan nilai aktual. MSE memberikan gambaran seberapa besar kesalahan prediksi model berikut:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

di mana y_i adalah nilai aktual dan \hat{y}_i adalah nilai prediksi model.

Interpretasi: Semakin kecil nilai MSE, semakin baik performa model dalam memprediksi kekuatan tekan beton.

2. Root Mean Squared Error (RMSE)

RMSE adalah akar kuadrat dari MSE, dan memberikan ukuran kesalahan dalam satuan yang sama dengan data asli. Hal tersebut membantu dalam memahami seberapa besar kesalahan prediksi secara lebih intuitif. Model persamaan sebagai berikut:

$$RMSE = \sqrt{MSE}$$

3. Mean Absolute Error (MAE)

MAE mengukur rata-rata nilai absolut dari selisih antara nilai yang diprediksi dan nilai actual. MAE memberikan ukuran kesalahan prediksi yang tidak terpengaruh oleh outliers. Model Persamaan sebagai berikut:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Interpretasi: MAE yang lebih kecil menunjukkan bahwa model memiliki kesalahan prediksi yang lebih rendah secara rata-rata.

4. Coefficient of Determination (R²):

R² mengukur proporsi variasi dalam data target yang dapat dijelaskan oleh model. Ini memberikan gambaran seberapa baik model menangkap hubungan antara fitur dan target. Model persamaan sebagai berikut:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

di mana \bar{y} adalah rata-rata nilai aktual.

Interpretasi: Nilai R² berkisar antara 0 hingga 1, di mana nilai yang lebih mendekati 1 menunjukkan bahwa model sangat baik dalam menjelaskan variabilitas data

5. Cross-Validation Score:

Cross-validation adalah teknik untuk mengevaluasi model dengan membaginya menjadi beberapa subset (folds) dan melatih serta menguji model pada subset yang berbeda. Ini memberikan indikasi tentang konsistensi performa model di berbagai subset data. Pada CVS menggunakan metode K-fold cross-validation. Metode ini umum di mana data dibagi menjadi k bagian, dan model dilatih serta diuji k kali, setiap kali dengan bagian data yang berbeda sebagai data uji.

Interpretasi: Skor rata-rata dari cross-validation memberikan gambaran umum tentang performa model pada data yang tidak terlihat selama pelatihan.

HASIL DAN PEMBAHASAN

Data yang digunakan dalam penelitian ini adalah data sekunder, sumber data berasal dari badan Standarisasi Nasional (BSN) tentang Spesifikasi beton Struktural disajikan data sebagai berikut:

Tabel 1. Datasets Mutu Beton (SNI)

Mutu Beton	Semen (KG)	Pasir (KG)	Kerikil (KG)	Air (Liter)
7.4 Mpa (K100)	247	869	999	215
9.8 Mpa (K125)	276	828	1012	215
12.2 Mpa (K150)	299	799	1017	215
14.5 Mpa (K175)	326	760	1029	215
16.9 Mpa (K200)	352	731	1031	215
19.3 Mpa (K225)	371	698	1047	215
21.7 Mpa (K250)	384	692	1039	215
24 Mpa (K275)	406	684	1026	215
26.4 Mpa (K300)	413	681	1021	215
28.8 Mpa (K325)	439	670	1006	215
31.2 Mpa (K350)	448	667	1000	215

Proses Input Datasets dengan Bahasa Pemrograman python

Pada dunia analisis data, langkah pertama yang krusial adalah mempersiapkan dan mengimpor data dengan benar. Sebelum kita dapat menggali wawasan dari dataset, kita perlu memastikan bahwa data tersebut diolah dengan cara yang efisien dan terstruktur. Dalam tutorial ini, kita akan fokus pada proses input datasets menggunakan Bahasa Pemrograman Python. Dengan Python, kita bisa memanfaatkan berbagai pustaka dan alat untuk mengelola data dengan mudah. Mari kita mulai dengan langkah awal yang sangat penting ini dan lihat bagaimana kita bisa mengimpor dataset ke dalam lingkungan kerja Python untuk analisis lebih lanjut. Berikut adalah kode pengumpulan datasets:

```
data = {  
'semen': [247, 276, 299, 326, 352, 371, 384, 406, 413, 439, 448],  
'pasir': [869, 828, 799, 760, 731, 698, 692, 684, 681, 670, 667],  
'kerikil': [999, 1012, 1017, 1029, 1031, 1047, 1039, 1026, 1021, 1006, 1000],  
'air': [215, 215, 215, 215, 215, 215, 215, 215, 215, 215, 215],  
'kuat_tekan': [7.4, 9.8, 12.2, 14.5, 16.9, 19.3, 21.7, 24, 26.4, 28.8, 31.2]  
}
```

Pada proses kedua penentuan data frame dari data yang telah dibangun. Setelah data diimpor, langkah berikutnya adalah membangun data frame menggunakan pustaka Pandas di Python. Data frame adalah struktur data tabular yang memudahkan manipulasi dan analisis data dengan cara yang efisien. Dengan menggunakan fungsi pd.DataFrame(), hal ini dapat mengonversi data mentah ke dalam format yang terstruktur, memungkinkan kita untuk dengan mudah mengakses dan memanipulasi data. Proses

ini memastikan bahwa data tersedia dalam format yang siap untuk analisis lebih lanjut, membantu kita mengelola dan memahami informasi dengan lebih baik.

Data Frame yang telah terbangun selanjutnya dilakukan reduksi data terkait penggunaan air. Hal ini dilakukan karena sesuai dengan Standar, air adalah molekul yang memiliki nilai tetap pada setiap atau masing – masing mutu beton. Selanjutnya dilakukan normalisasi fitur pada dataset dengan menggunakan “*StandardScaler*” dari pustaka “*sklearn.preprocessing*”. *StandardScaler()*: Ini adalah objek dari kelas *StandardScaler* yang digunakan untuk normalisasi fitur. *StandardScaler* mengubah data sehingga memiliki rata-rata 0 dan deviasi standar 1, yang membuat data lebih seragam dan memudahkan proses analisis atau pelatihan model. “*scaler.fit_transform(X)*” : Fungsi ini melakukan dua hal sekaligus: *fit(X)*: Menghitung rata-rata dan deviasi standar dari fitur dalam dataset X berdasarkan data yang diberikan. *transform(X)*: Mengubah data X dengan menggunakan rata-rata dan deviasi standar yang telah dihitung, menghasilkan data yang dinormalisasi. *X_scaled*: Ini adalah hasil dari proses normalisasi, di mana data fitur X telah dinormalisasi sehingga memiliki distribusi dengan rata-rata 0 dan deviasi standar 1. Proses ini penting dalam banyak algoritma pembelajaran mesin karena dapat meningkatkan kinerja model dengan memastikan bahwa semua fitur berada dalam skala yang sama.

Data – data yang telah di bentuk akan dilanjutkan dilakukan data latih menggunakan source code berikut : *X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)*. Membagi data menjadi data latih dan data uji merupakan tahap langkah krusial dalam pengembangan model machine learning. Data latih digunakan untuk melatih model, sedangkan data uji digunakan untuk mengevaluasi kinerjanya pada data yang belum pernah dilihat sebelumnya. Proses ini membantu mencegah overfitting, memastikan model dapat menggeneralisasi dengan baik, dan memberikan gambaran akurat tentang seberapa efektif model akan bekerja pada data baru di dunia nyata.

Generalisasi data menjadi keputusan dalam membuat model Neural Network. Beberapa *tuning* diaplikasikan menggunakan *hidden layer, activation, dan solver*. hal ini menjadi batas agar nilai dapat disesuaikan dengan mode MLP. Berikut source code pembuatan model dengan parameter iterasi dan grid:

```
model = GridSearchCV(MLPRegressor(max_iter=1000, random_state=42), parameter_grid, cv=5)
model.fit(X_train, y_train).
```

Model yang terbentuk selanjutnya dilakukan crossing dengan y prediksi untuk melakukan pengujian model (*X_test*). Hasil pengujian dilakukan evaluasi dengan fungsi mean squared error. Fungsi *mean_squared_error* dalam pustaka *sklearn.metrics* digunakan untuk mengukur seberapa baik model regresi dalam memprediksi nilai kontinu dengan menghitung rata-rata kuadrat selisih antara nilai prediksi dan nilai sebenarnya. Metrik ini membantu menilai akurasi model, di mana nilai MSE yang lebih rendah menunjukkan prediksi yang lebih akurat.

Hasil evaluasi model selanjutnya dilakukan uji kekuatan tekan menggunakan komposisi yang telah terbangun melalui datasets. Selanjutnya hasil kekuatan prediksi akan dilakukan optimasi sesuai dengan target yang telah ditetapkan kekuatannya. Jika kekuatan prediksi telah dioptimasi maka program akan melakukan rekomendasi sesuai dengan kekuatan yang di target berdasarkan komposisinya. berikut program evaluasi, prediksi, dan rekomendasinya :

```
# Fungsi untuk prediksi kekuatan tekan berdasarkan komposisi
def prediksi_kuat_tekan(composition):
    composition_scaled = scaler.transform([composition])
    return model.predict(composition_scaled)[0]
# Fungsi tujuan untuk optimasi
def tujuan_optimasi(composition, target_kuat_tekan):
    prediksi = prediksi_kuat_tekan(composition)
    return (prediksi - target_kuat_tekan) ** 2
```

```
#Fungsi untuk merekomendasikan komposisi berdasarkan kekuatan tekan target
def rekomendasi_komposisi(target_kuat_tekan):
# Batasan untuk komposisi: semen, pasir, dan kerikil tidak boleh negatif
batas = [(0, None), (0, None), (0, None)]
# Nilai awal (dapat diubah sesuai kebutuhan
nilai_awal = [300, 600, 1200]
# Menyelesaikan optimasi
hasil = minimize(tujuan_optimasi, nilai_awal, args=(target_kuat_tekan), bounds=batas) return
hasil.x
# Fungsi untuk input uji kubus dan rekomendasi komposisi
def input_uji_kubus(target_kuat_tekan):
# Komposisi default untuk perhitungan
rekomendasi = rekomendasi_komposisi(target_kuat_tekan)
print(f"Rekomendasi Komposisi untuk {target_kuat_tekan} MPa:")
print(f"Semen: {rekomendasi[0]:.2f} kg")
print(f"Pasir: {rekomendasi[1]:.2f} kg")
print(f"Kerikil: {rekomendasi[2]:.2f} kg")
print(f"Air: 215 L")
```

RUNNING PROGRAM

Hasil analisa dengan menggunakan running program data inputan menunjukkan bahwa program dapat menentukan seberapa kuat tekan beton yang di inginkan. Beberapa output yang di sarankan juga memiliki variasi dalam komposisi sehingga memiliki kekuatan sesuai dengan prediksi target. Berikut hasil prediksi dari program yang telah dibuat dengan menggunakan paramter kekuatan beton SNI.

Tabel 2. Hasil Komposisi Prediksi

Mutu Beton	Semen (KG)	Pasir (KG)	Kerikil (KG)	Air (Liter)
7.4 Mpa (K100)	321	612.15	1207.42	215
9.8 Mpa (K125)	327.78	615.54	1209.49	215
12.2 Mpa (K150)	333.83	618.93	1211.56	215
14.5 Mpa (K175)	339.64	622.17	1213.54	
16.9 Mpa (K200)	345.69	625.56	1215.61	215
19.3 Mpa (K225)	351.74	628.95	1217.68	215
21.7 Mpa (K250)	357.80	632.33	1219.74	215
24 Mpa (K275)	363.6	635.58	1221.73	215
26.4 Mpa (K300)	369.65	638.96	1223.80	215
28.8 Mpa (K325)	375.7	642.35	1225.86	215
31.2 Mpa (K350)	381.76	645.74	1227.93	215

Pada tabel diatas menunjukkan bahwa neural network dapat melakukan pembelajaran Machine Learning dari dataset yang telah di bangun dari SNI. Pada penelitian ini akan melakukan komparasi selisih nilai dari hasil komposisi, hal ini dilakukan karena pada kenyataannya seluruh komposisi mutu beton selalu terjadi koreksi meskipun telah sesuai dengan standar SNI untuk mencapai kuat tekan yang di targetkan. Berikut tabel komparasi dari setiap masing – masing material :

Tabel 3. Selisih Komposisi Program dengan SNI

Mutu Beton	Semen (KG)	Pasir (KG)	Kerikil (KG)	Air (Liter)
7.4 Mpa (K100)	30	30	21	0
9.8 Mpa (K125)	19	-26	20	0
12.2 Mpa (K150)	12	-23	19	0
14.5 Mpa (K175)	4	-18	18	0
16.9 Mpa (K200)	-2	-14	18	0
19.3 Mpa (K225)	-5	-10	16	0
21.7 Mpa (K250)	-7	-9	17	0
24 Mpa (K275)	-10	-7	19	0
26.4 Mpa (K300)	-10	-6	20	0
28.8 Mpa (K325)	-14	-4	22	0
31.2 Mpa (K350)	-15	-3	23	0

Hasil running program menunjukkan variatif dalam menemukan kuat tekan berdasarkan komposisinya. Nilai dari setiap komposisi variabel merupakan variansi dalam membuat sebuah betoon berukuran 1m³, dimana hasil menunjukkan bahwa ketika salah satu variabel bertambah naik maka ada variabel lain yang di turunkan untuk mencapai titik optimalnya (berlaku juga sebaliknya).

KESIMPULAN DAN SARAN

KESIMPULAN

Neural Network merupakan solusi praktis untuk menentukan pembelajaran mesin dari data data yang ada. Algoritma dan metode ini cukup signifikan dalam menentukan prediksi sesuai dengan data sebelumnya. Selain itu, pada kasus real banyak sekali praktisi dalam menentukan komposisi atau memproduksi beton sesuai dengan budget yang di inginkan. Sehingga dapat meminimalkan cost produksi sesuai ketersediaan barang yang ada dan dari penelitian ini maka dapat diambil beberapa kesimpulan penting sebagai berikut:

1. Neural network dapat melakukan pembelajaran dari dataset SNI dan memberikan prediksi yang akurat mengenai komposisi beton yang diperlukan untuk mencapai kekuatan tekan tertentu.
2. Terdapat variasi dalam komposisi yang direkomendasikan oleh model dibandingkan dengan data SNI, namun variasi tersebut menunjukkan fleksibilitas model dalam mencapai target kekuatan tekan.
3. Hasil menunjukkan bahwa peningkatan atau penurunan salah satu variabel komposisi akan mempengaruhi variabel lainnya untuk mencapai titik optimal, sesuai dengan prediksi target kekuatan beton

SARAN

Penelitian ini telah berhasil mengembangkan model prediksi kekuatan tekan beton berdasarkan komposisi material menggunakan metode neural network. Melalui analisis data yang komprehensif dan penerapan teknik machine learning, model ini mampu memberikan rekomendasi komposisi material yang optimal untuk mencapai kekuatan tekan yang diinginkan. Hasil prediksi menunjukkan adanya variasi komposisi yang signifikan, yang mencerminkan pentingnya penyesuaian material untuk mencapai performa beton yang optimal. Meskipun demikian, masih ada ruang untuk

peningkatan dalam penelitian ini melalui beberapa saran yang dapat meningkatkan akurasi model dan validitas hasil prediksi. Pada penelitian selanjutnya dapat disarankan beberapa hal sebagai berikut:

1. Penambahan Variabel Baru
 - a. Admixtures (Bahan Tambahan)
 - b. Jenis Semen
 - c. Gradasi Agregat
 - d. Kondisi Curing
2. Uji Fisik
 - a. Pengujian Sampel Beton
 - b. Uji Durabilitas
 - c. Uji Workability
3. Optimasi Model:
 - a. Hyperparameter Tuning
 - b. Cross-Validation
4. Integrasi Data Lapangan
 - a. Data Real-Time
 - b. Feedback Loop
5. Kolaborasi dengan Industri:
 - a. Kemitraan dengan Perusahaan Konstruksi
 - b. Pelatihan dan Penyuluhan

DAFTAR PUSTAKA

- ACI. (1994). Fly ash concrete mix design and the water-cement ratio law. *ACI Materials Journal*, 91(4). <https://doi.org/10.14359/4050>
- Abriantoro, A.P. (2023). Pengaruh fly ash sebagai substitusi semen terhadap durasi initial setting time, flowability dan kuat tekan umur 1 hari beton self-compacting concrete (SCC) dengan penambahan 0,15% citric acid. *Jurnal Kajian Teknik Sipil*, 7(2), 32–43. <https://doi.org/10.52447/jkts.v7i2.6852>
- Abriantoro, A. P., & Susilowati, E. (n.d.). Eksperimen substitusi serbuk kaca pada binder untuk meningkatkan kuat tekan beton self-compacting concrete (SCC) pada usia 1 hari. Unpublished.
- Argiz, C., Moragues, A., & Menéndez, E. (2018). Use of ground coal bottom ash as cement constituent in concretes exposed to chloride environments. *Journal of Cleaner Production*, 170, 25–33. <https://doi.org/10.1016/j.jclepro.2017.09.117>
- Bhanja, S., & Sengupta, B. (2002). Investigations on the compressive strength of silica fume concrete using statistical methods. *Cement and Concrete Research*, 32(9), 1391–1394. [https://doi.org/10.1016/S0008-8846\(02\)00787-1](https://doi.org/10.1016/S0008-8846(02)00787-1)
- Hasan, H. M. R., Rabby, A. S. A., Islam, M. T., & Hossain, S. A. (2019). Machine learning algorithm for student's performance prediction. In 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1–7). IEEE. <https://doi.org/10.1109/ICCCNT45670.2019.8944629>
- Kamath, M. V., Prashanth, S., Kumar, M., & Tantri, A. (2022). Machine-learning-algorithm to predict the high-performance concrete compressive strength using multiple data. *Journal of Engineering, Design and Technology*. <https://doi.org/10.1108/JEDT-11-2021-0637>
- Kim, T., Davis, J. M., Ley, M. T., Kang, S., & Amrollahi, P. (2018). Fly ash particle characterization for predicting concrete compressive strength. *Construction and Building Materials*, 165, 560–571. <https://doi.org/10.1016/j.conbuildmat.2018.01.059>

- Marcelino, P., de Lurdes Antunes, M., Fortunato, E., & Gomes, M. C. (2021). Machine learning approach for pavement performance prediction. *International Journal of Pavement Engineering*, 22(3), 341–354. <https://doi.org/10.1080/10298436.2019.1609673>
- Mostafaei, H., Badarloo, B., Chamasemani, N. F., Rostampour, M. A., & Lehner, P. (2023). Investigating the effects of concrete mix design on the environmental impacts of reinforced concrete structures. *Buildings*, 13(5), 1313. <https://doi.org/10.3390/buildings13051313>
- Naderpour, H., & Mirrashid, M. (2020). Estimating the compressive strength of eco-friendly concrete incorporating recycled coarse aggregate using neuro-fuzzy approach. *Journal of Cleaner Production*, 265, 121886. <https://doi.org/10.1016/j.jclepro.2020.121886>
- Nemirovsky, D., Arkose, T., Markovic, N., Nemirovsky, M., Unsal, O., & Cristal, A. (2017). A machine learning approach for performance prediction and scheduling on heterogeneous CPUs. In *2017 29th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)* (pp. 121–128). IEEE. <https://doi.org/10.1109/SBAC-PAD.2017.23>
- Pushkar, S. (2019). Life-cycle assessment of the substitution of sand with coal bottom ash in concrete: Two concrete design methods. *Applied Sciences**, 9(17), 3620. <https://doi.org/10.3390/app9173620>
- Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. **SN Computer Science**, 2(3), 160. <https://doi.org/10.1007/s42979-021-00592-x>
- T. Kim, J. M. Davis, M. T. Ley, S. Kang, and P. Amrollahi. (2018). Fly ash particle characterization for predicting concrete compressive strength. **Construction and Building Materials**, 165, 560–571. <https://doi.org/10.1016/j.conbuildmat.2018.01.059>