



ANALISIS PREDIKSI PENJUALAN PRODUK MENGUNAKAN ALGORITMA K-NEAREST NEIGHBOR REGRESSION

**Ratna Salkiawati^{1*}, Hendarman Lubis², Nurfiyah³,
Abdillah Syah Al Malik MK.⁴**

Program Studi Informatika¹, Program Studi Informatika²,
Program Studi Informatika³, Program Studi Informatika⁴
Fakultas Ilmu Komputer¹, Fakultas Ilmu Komputer²,
Fakultas Ilmu Komputer³, Fakultas Ilmu Komputer⁴
Universitas Bhayangkara Jakarta Raya¹,
Universitas Bhayangkara Jakarta Raya²,
Universitas Bhayangkara Jakarta Raya³,
Universitas Bhayangkara Jakarta Raya⁴,

*Correspondent Author: ratna_tind@dsn.ubharajaya.ac.id

Author Email: ratna_tind@dsn.ubharajaya.ac.id¹,
hendarman.lubis@dsn.ubharajaya.ac.id²,
nurfiyah@dsn.ubharajaya.ac.id³,
abdillah.syah17@mhs.ubharajaya.ac.id⁴

Received: October 12,2025. **Revised:** November 10,2025. **Accepted:**
November 11, 2025. **Issue Period:** Vol.9 No.2 (2025), Pp. 350-262

Abstrak: Toko Sembako merupakan usaha yang bergerak dibidang perdagangan, produk yang menyediakan produk sembako. Permasalahan yang ada yaitu belum adanya prediksi penjualan produk di masa mendatang berdasarkan data yang telah direkam sebelumnya. Prediksi ini untuk mempermudah serta memberitahu pemilik toko mengenai produk yang paling banyak dibeli konsumen. Untuk mengetahui prediksi penjualan produk digunakan metode K-Nearest Neighbor Regression, selanjutnya akan melewati tahapan Knowledge Discovery in Database (KDD). Hasil yang diperoleh penulis dari penelitian ini adalah nilai $k = 2$ mendapat RMSE 0.31318 untuk produk Beras pada bulan ke-11, $k = 2$ mendapat RMSE 0.29367 untuk produk Gula Pasir pada bulan ke-11 dan bulan ke-12, $k = 4$ mendapat RMSE 0.34002 untuk produk Minyak Goreng 2 Ltr pada bulan ke-11 dan bulan ke-12, $k = 3$ mendapat RMSE 0.31820 untuk produk Telur Ayam pada bulan ke-11, $k = 6$ mendapat RMSE 0.44437 untuk produk Tepung Terigu pada bulan ke-8. Seperti yang tertera dalam pedoman RMSE dapat disimpulkan bahwa seluruh model yang penulis uji memiliki satu kesalahan yang kecil yaitu 0,00 – 0,299 dan empat kesalahan sedang yaitu 0,30 – 0,559.



DOI: 10.52362/jisicom.v9i2.2132

Ciptaan disebarluaskan di bawah [Lisensi Creative Commons Atribusi 4.0 Internasional](https://creativecommons.org/licenses/by/4.0/).



Kata kunci: K-Nearest Neighbor Regression; Prediksi; Knowledge Discovery in Database; RMSE

Abstract: *The grocery store is a business engaged in trade, providing staple food products. The existing problem is the absence of sales prediction for future periods based on previously recorded data. This prediction aims to assist and inform the store owner about the products most frequently purchased by customers. To determine product sales predictions, the K-Nearest Neighbor Regression method is used, followed by the stages of Knowledge Discovery in Database (KDD). The results obtained from this research show that the value of $k = 2$ yields an RMSE of 0.31318 for rice products in the 11th month, $k = 2$ yields an RMSE of 0.29367 for granulated sugar products in the 11th and 12th months, $k = 4$ yields an RMSE of 0.34002 for 2-liter cooking oil products in the 11th and 12th months, $k = 3$ yields an RMSE of 0.31820 for chicken egg products in the 11th month, and $k = 6$ yields an RMSE of 0.44437 for wheat flour products in the 8th month. As stated in the RMSE guidelines, it can be concluded that all models tested by the author have one small error (0.00–0.299) and four moderate errors (0.30–0.559).*

Keywords: K-Nearest Neighbor Regression; Prediction; Knowledge Discovery in Databases; RMSE

I. PENDAHULUAN

Penjualan merupakan unsur yang penting dalam dunia usaha. Pemilik usaha tentunya memiliki tujuan jangka panjang agar usahanya dapat mengalami perkembangan di masa yang akan datang. Salah satunya dengan membuat prediksi atau peramalan. Prediksi atau peramalan digunakan untuk menemukan informasi dari data dalam jumlah yang sangat besar, sehingga diperlukan proses *data mining*. *Data mining* memiliki kemampuan untuk memberikan informasi yang dapat membantu dalam membuat keputusan yang berlandaskan pada data. Pada penelitian ini penulis menggunakan metode *K-Nearest Neighbor*.

Toko Sembako merupakan usaha yang bergerak dibidang perdagangan, yang menyediakan bahan pokok. Toko ini telah menerapkan sistem penjualan secara online dalam mempromosikan penjualan produknya. Data penjualan produk sembako dalam 3 tahun, menunjukkan peningkatan pada beberapa produk, tapi juga ada penurunan dalam beberapa produk. Hal ini membuat pemilik Toko Sembako sulit untuk memprediksi produk mana yang paling banyak dibeli, prediksi ini mempermudah serta memberitahu pemilik toko mengenai produk yang paling banyak dibeli oleh konsumen, dan juga belum bisa memprediksi penjualan produk di masa mendatang berdasarkan data yang telah direkam sebelumnya. Sehingga diperlukan proses *data mining* untuk mengolah data dalam skala besar secara efisien dan efektif, dan menggunakan metode *K-Nearest Neighbor* untuk memprediksi penjualan produk sembako yang paling banyak dibeli oleh konsumen.

Data mining adalah proses menemukan korelasi, pola, dan tren yang bermakna serta baru dengan menyaring sejumlah besar data yang tersimpan dalam repositori, menggunakan teknik pengenalan pola (pattern recognition) serta teknik statistik dan matematika. *Data mining* bertujuan untuk menemukan pola data yang bermanfaat dari sejumlah besar data. Dimana pada data mining menggunakan *machine learning* untuk teknik analisisnya. Teknik *machine learning* telah banyak digunakan di berbagai bidang ilmu pengetahuan, salah satunya *K-nearest neighbors (KNN)* merupakan salah satu metode *machine learning* yang sederhana. Pada dasarnya cara kerja algoritma *K-Nearest Neighbor (KNN)* adalah mencari jarak terdekat dengan k tetangga (*neighbor*) terdekat dalam data training dengan data yang akan diuji. Kemudian mengelompokkan data baru k dengan cara menghitung jarak data baru ke beberapa data/tetangga (*neighbor*) terdekat. Algoritma *K-Nearest Neighbor (KNN)* merupakan instead-based learning, dimana data training disimpan sehingga klasifikasi untuk record baru yang belum diklasifikasi dapat ditemukan dengan membandingkan kemiripan yang paling banyak dalam data training[1], [2], [3].



DOI: 10.52362/jisicom.v9i2.2132

Ciptaan disebarluaskan di bawah [Lisensi Creative Commons Atribusi 4.0 Internasional](https://creativecommons.org/licenses/by/4.0/).



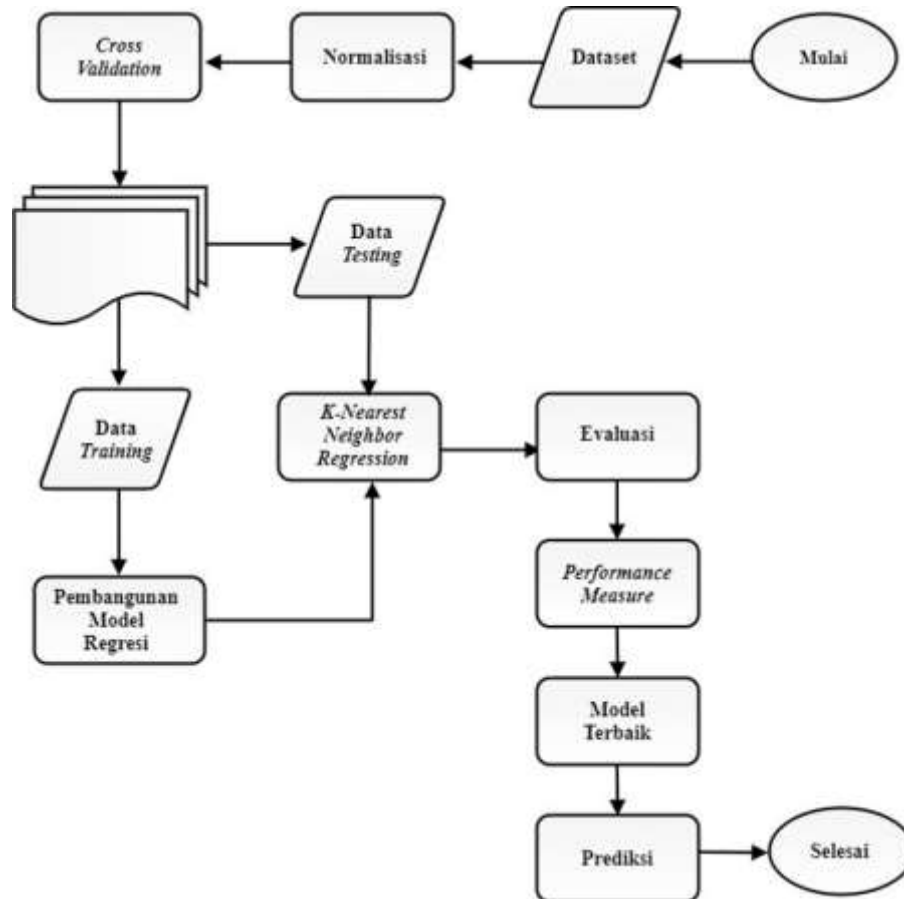
Penelitian yang dilakukan sebelumnya pada prediksi penjualan dengan pengujian menggunakan algoritma *K- Nearest Neighbor Regression* didapat nilai $k = 2, 3, 13$, dan 14 serta nilai RMSE antara $0,43-0,48$. Ini artinya model *KNN Regression* sudah cukup baik, tetapi belum sempurna dalam memprediksi penjualan[4]. Penelitian yang dilakukan[5] untuk memprediksi penjualan Donut dengan algoritma *K- Nearest Neighbor Regression* pada penelitian ini menunjukan prediksi penjualan terlaris pada produk Blackforest dengan $k = 12$ RMSE = 0.41876 , Brownies dengan $k = 4$ RMSE = 0.46398 , Coklat dengan $k = 9$ RMSE = 0.4827 , Donat dengan $k = 14$ RMSE = 0.47980 , Tart dengan $k = 12$ RMSE = 0.47381 , Triple Disert Box dengan $k = 13$ RMSE = 0.49336 . Karena semua nilai RMSE berada di antara $0,41 - 0,49$, maka tingkat kesalahan model termasuk sedang, hasilnya sudah cukup baik dan stabil. Penelitian untuk memprediksi harga saham menggunakan algoritma *K- Nearest Neighbor Regression* dengan empat variasi indikator yang berbeda, prediksi dengan menggunakan satu indikator memiliki hasil yang paling baik dengan nilai *root mean squared error* 169 dengan akurasi 98.8% dengan rasio perbandingan 70% data latih dan 30% data uji [6] Penelitian lainnya prediksi jumlah produksi coconut oil menggunakan *K-Nearest Neighbor* dan *Backward Elimination* didapatkan hasil penelitiannya model terbaik yang dilihat berdasarkan nilai error terkecil yaitu 0.111. Kemudian algoritma *k- Nearest Neighbor* menggunakan *Backward Elimination* dihasilkan model terbaik yang dilihat berdasarkan nilai error terkecil yaitu 0.109. Seleksi fitur yaitu *Backward Elimination* menghasilkan kinerja lebih baik[7]. Selanjutnya [8] melakukan prediksi laju pertumbuhan jumlah penduduk provinsi Kalimantan Selatan menggunakan metode *k-nearest neighbor regression* didapat hasil analisa data yang di gunakan di atas dapat disimpulkan hasil dari *Mean Squared Error (RMSE)* terbaik ada pada data label pada tahun 2012 dengan menggunakan data training 20% dan data testing 80% dengan memperoleh hasil Akurasi sebanyak 1.266 ± 0.000 lebih baik dari hasil yang lain setelah di lakukannya beberapa kali analisa. Selanjutnya [9] dengan penelitian Prediksi Harga Beras Premium dengan algoritma *K-Nearest Neighbor* didapat hasil penelitian prediksi terhadap harga beras pada tahun 2014 - 2019 dengan nilai RMSE 0,125 dan parameter $K = 2$ yang sudah dinormalisasi. Penelitian yang dilakukan oleh[10] identifikasi citra garis telapak tangan menggunakan metode GLCM dan *K-Nearest Neighbor* dihasilkan nilai akurasi sebesar 66.7%.

Dari penelitian terdahulu diatas, maka akan dilakukan penelitian dengan menerapkan algoritma *K-Nearest Neighbor Regression* untuk memprediksi produk sembako agar dapat mengetahui produk mana yang paling laris di bulan-bulan tertentu berdasarkan data penjualan sebelumnya, serta untuk mengukur tingkat akurasi model prediksi menggunakan metrik *RMSE (Root Mean Square Error)*.

II. METODE DAN MATERI

Berikut dibawah ini tahapan penelitian yang dilakukan:





Gambar 1. Tahapan Penelitian

Tahapan penelitian yang dilakukan dalam sistem prediksi penjualan produk sembako menggunakan metode *K-Nearest Neighbor* terdiri atas beberapa langkah yang saling berkaitan, dimulai dari [11], [12]:

1. tahapan awal merupakan inisialisasi proses penelitian yang dimulai dengan menyiapkan kebutuhan data serta menentukan tujuan penelitian, yaitu melakukan prediksi penjualan produk di masa mendatang.
2. dataset merupakan kumpulan data historis penjualan produk sembako yang digunakan sebagai bahan utama dalam proses analisis. Data ini diperoleh dari pencatatan transaksi penjualan di toko sembako dan digunakan untuk melatih serta menguji model prediksi. Dataset dibangun dari data penjualan mulai dari bulan ke-1 tahun ke-1 sampai dengan bulan ke-12 tahun ke -3.
3. tahap normalisasi data dilakukan dengan cara membuat data yang sudah ada menjadi nilai yang lebih kecil.
4. hasil dari normalisasi dilakukan *cross Validation*, data yang digunakan untuk training adalah data penjualan tahun ke-1 sampai tahun ke-3 sedangkan data yang digunakan untuk testing adalah data penjualan tahun ke 3[13]
5. setelah dilakukan pembagian data, sebagian data digunakan sebagai data training. Data ini digunakan untuk membangun model dan melatih algoritma agar mampu mengenali pola hubungan antara data input dan output.
6. pada tahap ini dilakukan proses pembangunan model regresi dengan menerapkan algoritma *K-Nearest Neighbor Regression*. Model dibentuk berdasarkan pola dari data training yang telah dinormalisasi.

7. data testing merupakan bagian dari dataset yang tidak digunakan dalam proses pelatihan model. Data ini digunakan untuk menguji kemampuan model dalam melakukan prediksi terhadap data baru yang belum pernah dilihat sebelumnya.
8. tahapan ini merupakan inti dari proses penelitian. Algoritma KNN Regression bekerja dengan mencari sejumlah tetangga terdekat (*k nearest neighbors*) dari data uji berdasarkan jarak tertentu (misalnya Euclidean distance), kemudian menghitung rata-rata nilai target dari tetangga-tetangga tersebut untuk menghasilkan nilai prediksi.
9. setelah model menghasilkan prediksi, dilakukan proses evaluasi untuk menilai kinerja model dengan membandingkan hasil prediksi terhadap nilai aktual.
10. evaluasi kinerja model diukur menggunakan metrik *Root Mean Square Error (RMSE)*. Nilai RMSE menunjukkan tingkat kesalahan prediksi; semakin kecil nilainya, semakin baik akurasi model yang dihasilkan.
11. model dengan nilai kesalahan terkecil (RMSE terendah) dipilih sebagai model terbaik.
12. pada tahap ini, model terbaik digunakan untuk melakukan prediksi penjualan di masa mendatang, seperti memprediksi penjualan pada tahun keempat berdasarkan data tahun sebelumnya.
13. tahap terakhir merupakan akhir dari proses penelitian, di mana hasil prediksi dan analisis model disajikan sebagai keluaran akhir yang dapat digunakan oleh pemilik toko untuk pengambilan keputusan terkait stok dan strategi penjualan sembako.

III. PEMBAHASAN DAN HASIL

3.1. Pengolahan data penelitian

Pada pembahasan dan hasil ini, akan dibahas mulai dari dataset penelitian sampai mendapatkan hasil prediksi produk terlaris dari penjualan di Toko Sembako Berkah dengan metode *K-Nearest Neighbor Regression*. Dataset penelitian ini merupakan data penjualan produk yang tersedia dari tahun 1-3. Dimana nanti dataset penjualan ini akan di bagi menjadi data training dan data testing. Berikut ini adalah data penjualan produk sembako:

Tabel 1. Data Penjualan Produk Sembako

No	Produk	Total Penjualan		
		Tahun 1	Tahun 2	Tahun 3
1	Beras	3599	6612	7345
2	Gula Pasir	1724	2328	2973
3	Minyak Goreng	1792	3740	3149
4	Telur Ayam	3704	5191	6061
5	Tepung Terigu	1721	3058	3171

Tabel 1 merupakan data penjualan setiap produk dalam 3 tahun. Dari data penjualan produk sembako selama 3 tahun dapat terlihat adanya kenaikan penjualan dari setiap produk.

3.2. Pembentukan dataset

Data training akan dipisahkan menjadi data input dan data target, data input merupakan data penjualan dari bulan 1-12 pada tahun 1-3, sedangkan data target merupakan data penjualan pada bulan ke-13. Data testing



DOI: 10.52362/jisicom.v9i2.2132

Ciptaan disebarluaskan di bawah [Lisensi Creative Commons Atribusi 4.0 Internasional](https://creativecommons.org/licenses/by/4.0/).



merupakan data penjualan pada bulan 1-12 di tahun ke-3 untuk menghasilkan prediksi penjualan produk tahun ke 4. Berikut ini pembagian dataset training dan dataset testing untuk produk sembako beras:

Tabel 2. Dataset training Beras

Data Input												Data Target
373	239	327	293	313	291	312	287	302	310	225	327	600
239	327	293	313	291	312	287	302	310	225	327	600	612
327	293	313	291	312	287	302	310	225	327	600	612	616
293	313	291	312	287	302	310	225	327	600	612	616	593
313	291	312	287	302	310	225	327	600	612	616	593	596
291	312	287	302	310	225	327	600	612	616	593	596	537
312	287	302	310	225	327	600	612	616	593	596	537	478
287	302	310	225	327	600	612	616	593	596	537	478	459
302	310	225	327	600	612	616	593	596	537	478	459	455
310	225	327	600	612	616	593	596	537	478	459	455	489
225	327	600	612	616	593	596	537	478	459	455	489	570
327	600	612	616	593	596	537	478	459	455	489	570	607
600	612	616	593	596	537	478	459	455	489	570	607	607
612	616	593	596	537	478	459	455	489	570	607	607	610
616	593	596	537	478	459	455	489	570	607	607	610	615
593	596	537	478	459	455	489	570	607	607	610	615	599
596	537	478	459	455	489	570	607	607	610	615	599	601
537	478	459	455	489	570	607	607	610	615	599	601	612
478	459	455	489	570	607	607	610	615	599	601	612	597
459	455	489	570	607	607	610	615	599	601	612	597	610
455	489	570	607	607	610	615	599	601	612	597	610	602
489	570	607	607	610	615	599	601	612	597	610	602	629
570	607	607	610	615	599	601	612	597	610	602	629	612
607	607	610	615	599	601	612	597	610	602	629	612	651

Tabel 3. Dataset testing Beras

Data Input											
612	616	593	596	537	478	459	455	489	570	607	607
616	593	596	537	478	459	455	489	570	607	607	610
593	596	537	478	459	455	489	570	607	607	610	615
596	537	478	459	455	489	570	607	607	610	615	599
537	478	459	455	489	570	607	607	610	615	599	601
478	459	455	489	570	607	607	610	615	599	601	612
459	455	489	570	607	607	610	615	599	601	612	597





455	489	570	607	607	610	615	599	601	612	597	610
489	570	607	607	610	615	599	601	612	597	610	602
570	607	607	610	615	599	601	612	597	610	602	629
607	607	610	615	599	601	612	597	610	602	629	612
607	610	615	599	601	612	597	610	602	629	612	651

3.3. Normalisasi Dataset

Proses normalisasi dilakukan dengan membuat data yang ada menjadi nilai yang lebih kecil. Data yang telah diolah akan dinormalisasi untuk menjadikan data menjadi jarak [0,1] yang berarti nilai minimal data yaitu 0, dan nilai maksimal data yaitu 1. Kemudian data diantara minimal dan maksimal disesuaikan jaraknya. Berikut ini merupakan implementasi pada program:

```
from sklearn.preprocessing import MinMaxScaler
# default range 0-1
mmScaler = MinMaxScaler()
dataset_norm = mmScaler.fit_transform(df)
pd.DataFrame(dataset_norm)
```

Gambar 2. Script Normalisasi

Hasil normalisasi yang diimplementasikan sebagai berikut untuk produk sembako beras:





0	0.378517	0.035806	0.260870	0.173913	0.225064	0.168798	0.222506	0.158568	0.196931	0.217391	0.000000	0.000000	0.739796
1	0.035806	0.260870	0.173913	0.225064	0.168798	0.222506	0.158568	0.196931	0.217391	0.000000	0.252475	0.903974	0.801020
2	0.260870	0.173913	0.225064	0.168798	0.222506	0.158568	0.196931	0.217391	0.000000	0.260870	0.928218	0.943709	0.821429
3	0.173913	0.225064	0.168798	0.222506	0.158568	0.196931	0.217391	0.000000	0.260870	0.959079	0.957921	0.956954	0.704082
4	0.225064	0.168798	0.222506	0.158568	0.196931	0.217391	0.000000	0.260870	0.959079	0.989770	0.967822	0.880795	0.719388
5	0.168798	0.222506	0.158568	0.196931	0.217391	0.000000	0.260870	0.959079	0.989770	1.000000	0.910891	0.890728	0.418367
6	0.222506	0.158568	0.196931	0.217391	0.000000	0.260870	0.959079	0.989770	1.000000	0.941176	0.918317	0.695364	0.117347
7	0.158568	0.196931	0.217391	0.000000	0.260870	0.959079	0.989770	1.000000	0.941176	0.948849	0.772277	0.500000	0.020408
8	0.196931	0.217391	0.000000	0.260870	0.959079	0.989770	1.000000	0.941176	0.948849	0.797954	0.626238	0.437086	0.000000
9	0.217391	0.000000	0.260870	0.959079	0.989770	1.000000	0.941176	0.948849	0.797954	0.647059	0.579208	0.423841	0.173469
10	0.000000	0.260870	0.959079	0.989770	1.000000	0.941176	0.948849	0.797954	0.647059	0.598465	0.569307	0.536424	0.586735
11	0.260870	0.959079	0.989770	1.000000	0.941176	0.948849	0.797954	0.647059	0.598465	0.588235	0.653465	0.804636	0.775510
12	0.959079	0.989770	1.000000	0.941176	0.948849	0.797954	0.647059	0.598465	0.588235	0.675192	0.853960	0.927152	0.775510
13	0.989770	1.000000	0.941176	0.948849	0.797954	0.647059	0.598465	0.588235	0.675192	0.882353	0.945545	0.927152	0.790816
14	1.000000	0.941176	0.948849	0.797954	0.647059	0.598465	0.588235	0.675192	0.882353	0.976982	0.945545	0.937086	0.816327
15	0.941176	0.948849	0.797954	0.647059	0.598465	0.588235	0.675192	0.882353	0.976982	0.976982	0.952970	0.953642	0.734694
16	0.948849	0.797954	0.647059	0.598465	0.588235	0.675192	0.882353	0.976982	0.976982	0.984655	0.965347	0.900662	0.744898
17	0.797954	0.647059	0.598465	0.588235	0.675192	0.882353	0.976982	0.976982	0.984655	0.997442	0.925743	0.907285	0.801020
18	0.647059	0.598465	0.588235	0.675192	0.882353	0.976982	0.976982	0.984655	0.997442	0.956522	0.930693	0.943709	0.724490
19	0.598465	0.588235	0.675192	0.882353	0.976982	0.976982	0.984655	0.997442	0.956522	0.961637	0.957921	0.894040	0.790816
20	0.588235	0.675192	0.882353	0.976982	0.976982	0.984655	0.997442	0.956522	0.961637	0.989770	0.920792	0.937086	0.750000
21	0.675192	0.882353	0.976982	0.976982	0.984655	0.997442	0.956522	0.961637	0.989770	0.951407	0.952970	0.910596	0.887755
22	0.882353	0.976982	0.976982	0.984655	0.997442	0.956522	0.961637	0.989770	0.951407	0.984655	0.933168	1.000000	0.801020
23	0.976982	0.976982	0.984655	0.997442	0.956522	0.961637	0.989770	0.951407	0.984655	0.964194	1.000000	0.943709	1.000000

Gambar 3. Hasil normalisasi untuk produk beras

Setelah data dinormalisasi, selanjutnya akan dilakukan tahap kedua pembagian dataset kembali. Variabel X dipakai untuk data input dengan data dimulai dari index ke-0 sampai dengan ke-11. Untuk variabel Y dipakai untuk target dengan data index ke-12. Berikut ini merupakan implementasi pada program :

```
X = dataset_norm[:,[0,1,2,3,4,5,6,7,8,9,10,11]]
y_norm = dataset_norm[:, [-1]]
y = y_norm.reshape(24)
pd.DataFrame(y)
```

Gambar 4. Script Pembagian Data Input Dan Data Target

Berikut ini hasil implementasinya:



DOI: 10.52362/jisicom.v9i2.2132

Ciptaan disebarluaskan di bawah [Lisensi Creative Commons Atribusi 4.0 Internasional](https://creativecommons.org/licenses/by/4.0/).



0	0.975155	1.000000	0.86250	0.88125	0.51250	0.14375	0.02500	0.00000	0.000000	0.000000	0.31250	0.185185
1	1.000000	0.857143	0.88125	0.51250	0.14375	0.02500	0.00000	0.21250	0.642857	0.627119	0.31250	0.240741
2	0.857143	0.875776	0.51250	0.14375	0.02500	0.00000	0.21250	0.71875	0.936508	0.627119	0.40625	0.333333
3	0.875776	0.509317	0.14375	0.02500	0.00000	0.21250	0.71875	0.95000	0.936508	0.677966	0.56250	0.037037
4	0.509317	0.142857	0.02500	0.00000	0.21250	0.71875	0.95000	0.95000	0.960317	0.762712	0.06250	0.074074
5	0.142857	0.024845	0.00000	0.21250	0.71875	0.95000	0.95000	0.96875	1.000000	0.491525	0.12500	0.277778
6	0.024845	0.000000	0.21250	0.71875	0.95000	0.95000	0.96875	1.00000	0.873016	0.525424	0.46875	0.000000
7	0.000000	0.211180	0.71875	0.95000	0.95000	0.96875	1.00000	0.90000	0.888889	0.711864	0.00000	0.240741
8	0.211180	0.714286	0.95000	0.95000	0.96875	1.00000	0.90000	0.91250	0.976190	0.457627	0.40625	0.092593
9	0.714286	0.944099	0.95000	0.96875	1.00000	0.90000	0.91250	0.98125	0.857143	0.677966	0.15625	0.592593
10	0.944099	0.944099	0.96875	1.00000	0.90000	0.91250	0.98125	0.88750	0.960317	0.542373	1.00000	0.277778
11	0.944099	0.962733	1.00000	0.90000	0.91250	0.98125	0.88750	0.96875	0.896825	1.000000	0.46875	1.000000

Gambar 5. Hasil normalisasi tahap kedua pada dataset untuk produk beras

3.4. Implementasi *K-Fold Cross Validation*

Pada tahap ini *K-Fold Cross Validation* dipakai untuk membagi data dan juga melakukan validasi data. Hasil split yang dikerjakan *K-Fold Cross Validation* adalah sebanyak n. Dimana pada penelitian ini nilai n yang digunakan yaitu n = 10. Berikut ini adalah hasilnya:

```

TRAIN: [ 0 1 2 4 5 6 7 8 9 10 11 12 14 15 16 17 19 20 21 22 23] TEST: [ 3 13 18]
TRAIN: [ 0 1 2 3 4 5 6 7 8 9 10 11 12 13 15 16 18 19 21 22 23] TEST: [14 17 20]
TRAIN: [ 0 1 3 5 6 7 8 9 11 12 13 14 15 16 17 18 19 20 21 22 23] TEST: [ 2 4 10]
TRAIN: [ 0 1 2 3 4 5 8 9 10 11 12 13 14 15 16 17 18 20 21 22 23] TEST: [ 6 7 19]
TRAIN: [ 0 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 22 23] TEST: [ 1 21]
TRAIN: [ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 17 18 19 20 21 22 23] TEST: [ 0 16]
TRAIN: [ 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 16 17 18 19 20 21 22] TEST: [15 23]
TRAIN: [ 0 1 2 3 4 5 6 7 8 10 11 12 13 14 15 16 17 18 19 20 21 23] TEST: [ 9 22]
TRAIN: [ 0 1 2 3 4 5 6 7 9 10 11 13 14 15 16 17 18 19 20 21 22 23] TEST: [ 8 12]
TRAIN: [ 0 1 2 3 4 6 7 8 9 10 12 13 14 15 16 17 18 19 20 21 22 23] TEST: [ 5 11]

```

Gambar 6. Hasil Implementasi *K-Fold Cross Validation* untuk produk beras



```
-----
Index TRAIN: [ 0 1 2 4 5 6 7 8 9 10 11 12 14 15 16 17 19 20 21 22 23] Index TEST: [ 3 13 18]
X_TRAIN:
[[0.37851662 0.03580563 0.26086957 0.17391304 0.22506394 0.16879795
 0.22250639 0.15856777 0.19693095 0.2173913 0.
 0.03580563 0.26086957 0.17391304 0.22506394 0.16879795 0.22250639
 0.15856777 0.19693095 0.2173913 0.
 0.25247525 0.90397351]
[0.26086957 0.17391304 0.22506394 0.16879795 0.22250639 0.15856777
 0.19693095 0.2173913 0.
 0.26086957 0.92821782 0.94370861]
[0.22506394 0.16879795 0.22250639 0.15856777 0.19693095 0.2173913
 0.
 0.26086957 0.95907928 0.98976982 0.96782178 0.8807947 ]
[0.16879795 0.22250639 0.15856777 0.19693095 0.2173913 0.
 0.26086957 0.95907928 0.98976982 1.
 0.91089109 0.89072848]
[0.22250639 0.15856777 0.19693095 0.2173913 0.
 0.26086957
 0.95907928 0.98976982 1.
 0.94117647 0.91831683 0.69536424]
[0.15856777 0.19693095 0.2173913 0.
 0.26086957 0.95907928
 0.98976982 1.
 0.94117647 0.9488491 0.77227723 0.5 ]
[0.19693095 0.2173913 0.
 0.26086957 0.95907928 0.98976982
 1.
 0.94117647 0.9488491 0.79795396 0.62623762 0.43708609]
[0.2173913 0.
 0.26086957 0.95907928 0.98976982 1.
 0.94117647 0.9488491 0.79795396 0.64705882 0.57920792 0.42384106]
[0.
 0.26086957 0.95907928 0.98976982 1.
 0.94117647
 0.9488491 0.79795396 0.64705882 0.59846547 0.56930693 0.53642384]
[0.26086957 0.95907928 0.98976982 1.
 0.94117647 0.9488491
 0.79795396 0.64705882 0.59846547 0.58823529 0.65346535 0.80463576]
[0.95907928 0.98976982 1.
 0.94117647 0.9488491 0.79795396
 0.64705882 0.59846547 0.58823529 0.67519182 0.85396084 0.92715232]
[1.
 0.94117647 0.9488491 0.79795396 0.64705882 0.59846547
 0.58823529 0.67519182 0.88235294 0.9769821 0.94554455 0.93708609]
[0.94117647 0.9488491 0.79795396 0.64705882 0.59846547 0.58823529
 0.67519182 0.88235294 0.9769821 0.9769821 0.9529703 0.95364238]
[0.9488491 0.79795396 0.64705882 0.59846547 0.58823529 0.67519182
 0.88235294 0.9769821 0.9769821 0.98465473 0.96534653 0.90066225]
[0.79795396 0.64705882 0.59846547 0.58823529 0.67519182 0.88235294
 0.9769821 0.9769821 0.98465473 0.99744246 0.92574257 0.90728477]
[0.59846547 0.58823529 0.67519182 0.88235294 0.9769821 0.9769821
 0.98465473 0.99744246 0.95652174 0.96163683 0.95792079 0.89403974]
[0.58823529 0.67519182 0.88235294 0.9769821 0.9769821 0.98465473
 0.99744246 0.95652174 0.96163683 0.98976982 0.92079208 0.93708609]
[0.67519182 0.88235294 0.9769821 0.9769821 0.98465473 0.99744246
 0.95652174 0.96163683 0.98976982 0.95140665 0.9529703 0.91059603]
[0.88235294 0.9769821 0.9769821 0.98465473 0.99744246 0.95652174
 0.96163683 0.98976982 0.95140665 0.98465473 0.93316832 1. ]
[0.9769821 0.9769821 0.98465473 0.99744246 0.95652174 0.96163683
 0.98976982 0.95140665 0.98465473 0.96419437 1.
 0.94370861]]
X_Test:
[[0.17391304 0.22506394 0.16879795 0.22250639 0.15856777 0.19693095
 0.2173913 0.
 0.26086957 0.95907928 0.95792079 0.95695364]
[0.98976982 1.
 0.94117647 0.9488491 0.79795396 0.64705882
 0.59846547 0.58823529 0.67519182 0.88235294 0.94554455 0.92715232]
[0.64705882 0.59846547 0.58823529 0.67519182 0.88235294 0.9769821
 0.9769821 0.98465473 0.99744246 0.95652174 0.93069307 0.94370861]]
Y_Train:
[[0.73979592 0.80102041 0.82142857 0.71938776 0.41836735 0.11734694
 0.02040816 0.
 0.17346939 0.58673469 0.7755102 0.7755102
 0.81632653 0.73469388 0.74489796 0.80102041 0.79081633 0.75
 0.8877551 0.80102041 1.
 ]
Y_Test: [0.70408163 0.79081633 0.7244898 ]
-----
```

Gambar 7. Hasil Implementasi split *K-Fold Cross Validation* untuk produk beras

Pada gambar 6 hasil tersebut adalah hasil split pada iterasi pertama dimana menampilkan data training x, test x, dan data training y lalu data test y.

3.5. Implementasi *K-Nearest Neighbor*

Pada tahapan ini model *K-Nearest Neighbor* Regression dibuat dengan nilai k nya yaitu 3 untuk sampel pengujian model yang akan dipakai pada evaluasi dengan *Cross Validation*. Berikut ini merupakan *script* nya:



```
## membangun model
model = KNeighborsRegressor(n_neighbors=3, metric='euclidean')
model

KNeighborsRegressor(algorithm='auto', leaf_size=30, metric='euclidean',
metric_params=None, n_jobs=None, n_neighbors=3, p=2,
weights='uniform')
```

Gambar 8. Implementasi K-Nearest Neighbor dan Cross Validation

3.6. Evaluasi

Evaluasi dilakukan untuk menguji model dengan menggunakan *K-Fold Cross Validation* untuk tiap nilai k dari K-Nearest Neighbor dengan range 1-15. Hasil dari keakuratan model prediksi dipengaruhi nilai k tersebut. Berikut ini hasil validasi:

```
RMSE value for k= 1 is: 0.34453952312209013
RMSE value for k= 2 is: 0.31318801811991076
RMSE value for k= 3 is: 0.3297855415286737
RMSE value for k= 4 is: 0.3407549641285234
RMSE value for k= 5 is: 0.37375644148431797
RMSE value for k= 6 is: 0.39956396301996217
RMSE value for k= 7 is: 0.4064889127043978
RMSE value for k= 8 is: 0.4172232676901432
RMSE value for k= 9 is: 0.4126793341133438
RMSE value for k= 10 is: 0.41450117549869014
RMSE value for k= 11 is: 0.4189301537449404
RMSE value for k= 12 is: 0.41715957239830587
RMSE value for k= 13 is: 0.42533884666764576
RMSE value for k= 14 is: 0.4456830509725474
```

Gambar 9. Nilai RMSE tiap k

Nilai tersebut menampilkan detail pada tiap k dalam range 1-15. Proses pengujian dilakukan setelah model dihasilkan dari tahapan proses training sebelumnya. Pengujian dilakukan untuk mengetahui akurasi dari implementasi K-Nearest Neighbor.

3.7. Model Terbaik dan Prediksi Produk

Setelah didapat nilai error terkecil, maka model yang dibentuk akan memakai nilai k dari hasil error terkecil tersebut.

	Bln 1	Bln 2	Bln 3	Bln 4	Bln 5	Bln 6	Bln 7	Bln 8	Bln 9	Bln 10	Bln 11	Bln 12	prediksi
0	612	616	593	596	537	478	459	455	489	570	607	607	0.772109
1	616	593	596	537	478	459	455	489	570	607	607	610	0.782313
2	593	596	537	478	459	455	489	570	607	607	610	615	0.624150
3	596	537	478	459	455	489	570	607	607	610	615	599	0.185374
4	537	478	459	455	489	570	607	607	610	615	599	601	0.045918
5	478	459	455	489	570	607	607	610	615	599	601	612	0.064626
6	459	455	489	570	607	607	610	615	599	601	612	597	0.253401
7	455	489	570	607	607	610	615	599	601	612	597	610	0.253401
8	489	570	607	607	610	615	599	601	612	597	610	602	0.511905
9	570	607	607	610	615	599	601	612	597	610	602	629	0.821429
10	607	607	610	615	599	601	612	597	610	602	629	612	0.896259
11	607	610	615	599	601	612	597	610	602	629	612	651	0.896259

Gambar 10. Hasil Prediksi penjualan produk beras

Nilai prediksi tersebut adalah ukuran skala dari normalisasi dengan nilai teringinya sebagai nilai penjualan yang paling laris. Berikut ini tabel hasil prediksi produk terlaris:

Tabel 4. Hasil Prediksi setiap produk

No	Produk	Prediksi
1	Beras	Bulan ke-11
2	Gula Pasir	Bulan ke-11 dan ke 12
3	Minyak Goreng	Bulan ke-11 dan ke 12
4	Telur Ayam	Bulan ke-11
5	Tepung Terigu	Bulan ke-8

IV. KESIMPULAN

Hasil penelitian dengan menerapkan metode *K-Nearest Neighbor Regression* untuk memprediksi penjualan produk sembako pada Toko Sembako didapatkan prediksi produk terlaris adalah Beras pada bulan ke-11, untuk Gula Pasir pada bulan ke-11 dan bulan ke-12, Minyak Goreng 2 Ltr pada bulan ke-11 dan bulan ke-12, Telur Ayam pada bulan ke-11, dan Tepung Terigu pada bulan ke-8. Dengan perolehan nilai k yang paling optimal dari rentang 1-15 pada setiap produk yaitu nilai k = 2, RMSE = 0.31318 untuk produk Beras ; nilai k = 2, RMSE = 0.29367 untuk produk Gula Pasir ; nilai k = 4, RMSE = 0.34002 untuk produk Minyak Goreng 2 Ltr; nilai k = 3, RMSE = 0.31820 untuk produk Telur Ayam ; nilai k = 6, RMSE = 0.44437 untuk produk Tepung Terigu . Berdasarkan pedoman interpretasi RMSE dapat disimpulkan bahwa tingkat kesalahan seluruh model yang penulis uji mempunyai satu kesalahan kecil dikarenakan nilai RMSE yang didapatkan berada dalam range 0,00 – 0,299 dan empat kesalahan sedang dikarenakan nilai RMSE yang didapatkan berada dalam range 0,30 – 0,599. Penelitian lebih lanjut bisa dilakukan dengan jumlah data yang lebih banyak dapat agar bisa menghasilkan training yang bervariasi dan mendapat hasil error yang lebih kecil.



DOI: 10.52362/jisicom.v9i2.2132

Ciptaan disebarluaskan di bawah [Lisensi Creative Commons Atribusi 4.0 Internasional](https://creativecommons.org/licenses/by/4.0/).



e-ISSN : 2597-3673 (Online) , p-ISSN : 2579-5201 (Printed)

Vol.9 No.2 (December 2025)

Journal of Information System, Informatics and Computing

Website/URL: <http://journal.stmikjayakarta.ac.id/index.php/jisicom>

Email: jisicom@stmikjayakarta.ac.id , jisicom2017@gmail.com

REFERENSI

- [1] V. Kotu and B. Deshpande, *Predictive analytics and data mining : concepts and practice with RapidMiner*. 225 Wyman Street, Waltham, MA 02451, USA: Morgan Kaufmann is an imprint of Elsevier, 2015.
- [2] Nong, Ye, *Data mining : theories, algorithms, and examples*. CRC Press, 2014.
- [3] Z. Zhang, "Introduction to machine learning: K-nearest neighbors," *Ann Transl Med*, vol. 4, no. 11, Jun. 2016, doi: 10.21037/atm.2016.03.37.
- [4] R. Salkiawati, H. Lubis, F. Sapril Telaumbanua, and F. Ilmu Komputer, "JSI (Jurnal Sistem Informasi) Universitas Dirgantara Marsekal Suryadarma Prediksi Penjualan Produk Sepatu dengan Menggunakan Algoritma K-Nearest Neighbor Regression dan Cross Validation," vol. 12, no. 1, pp. 48–57, Jan. 2025.
- [5] R. Salkiawati, H. Lubis, I. Nur El Khozy, P. Studi Informatika, F. Ilmu Komputer, and U. Bhayangkara Jakarta Raya, "K-Nearest Neighbor Regression untuk Prediksi Penjualan Donut," *Journal of Information System, Informatics and Computing Issue Period*, vol. 9, no. 1, pp. 180–190, 2025, doi: 10.52362/jisicom.v9i1.1964.
- [6] D. Nurfauzan and T. Fatimah, "Implementasi Algoritma K-Nearest Neighbors Regression Dalam Memprediksi Harga Saham," 2022. [Online]. Available: <https://senafit.budiluhur.ac.id/index.php/>
- [7] I. Colanus and R. Drajana, "Prediksi Jumlah Produksi Coconut Oil Menggunakan k-Nearest Neighbor dan Backward Elimination."
- [8] F. Ekawati, "Prediksi Laju Pertumbuhan Jumlah Penduduk Provinsi Kalimantan Selatan menggunakan metode K-Nearest Neighbor Regression," 2022.
- [9] Y. Mukhlisin, M. Imrona, and D. T. Murdiansyah, "Prediksi Harga Beras Premium dengan Metode Algoritma K-Nearest Neighbor."
- [10] L. H. S. R. Nurfiyah, "Identifikasi Citra Garis Telapak Tangan Menggunakan Metode GLCM & K-NN," *Jurnal Manajemen Informatika Jayakarta*, vol. 4, no. 3, pp. 251–257, Jul. 2024.
- [11] Larose, *Discovering Knowledge in Data: An Introduction to Data Mining*. New Jersey: Published by John Wiley & Sons, Inc., Hoboken, 2005.
- [12] Jiawei Han, Micheline Kamber, and Jian Pei, *Data Mining Third Edition*, Third. 225 Wyman Street, Waltham, MA 02451, USA: Morgan Kaufmann Publishers is an imprint of Elsevier, 2012.
- [13] D. Berrar, "Cross-validation," in *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, vol. 1–3, Elsevier, 2018, pp. 542–545. doi: 10.1016/B978-0-12-809633-8.20349-X.



DOI: 10.52362/jisicom.v9i2.2132

Ciptaan disebarluaskan di bawah [Lisensi Creative Commons Atribusi 4.0 Internasional](https://creativecommons.org/licenses/by/4.0/).