

Tugas MK. DATA MINING DAN INFORMATION RETRIEVAL TA 2024-2025

KLASIFIKASI

K - NEAREST NEIGHBOR

Disusun untuk memenuhi salah satu tugas mata kuliah IFB- 307 yang diberikan oleh: Jasman Pardede,
Dr., S.Si., M.T.



Disusun oleh :

Deden Fahrul Roziqin	152022182
Muhammad Yazid	152022192
Hamzah Sabahedthin	152022204
Budi Amin	152022213

Kelas DD

INSTITUT TEKNOLOGI NASIONAL BANDUNG
FAKULTAS TEKNOLOGI INDUSTRI INFORMATIKA
2024/2025

Kata Pengantar

Puji syukur kami panjatkan kehadirat Allah SWT yang Maha Kuasa yang telah melimpahkan karunia dan rahmat-Nya sehingga kami dapat menyusun laporan tugas Klasifikasi dengan menggunakan k-nearest neighbor untuk memenuhi tugas mata kuliah Data Mining dan Information Retrieval.

Dalam penyusunan laporan ini, tidak lepas dari adanya dukungan dan bantuan dari berbagai pihak. Oleh karena itu, kami mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan dan bantuan, terutama kepada:

1. Bapak Jasman Pardede, Dr., S.Si., M.T. selaku Dosen mata kuliah Data Mining Dan Information Retrieval.
2. Dan teman-teman semua yang telah membantu dalam memberikan saran dalam penyusunan laporan akhir ini.

Kami sebagai penulis dan penyusun menyadari bahwa laporan ini masih jauh dari kata sempurna.

Oleh karena itu kritik dan saran yang bersifat membangun dari semua pihak sangat diharapkan. Agar kedepannya laporan ini dapat menjadi lebih baik lagi. Kami berharap, semoga laporan ini dapat bermanfaat baik bagi kami pada khususnya dan para pembaca pada umumnya.

Bandung, 14 Oktober 2024

Penyusun

DAFTAR ISI

Kata Pengantar.....	1
BAB I PENDAHULUAN.....	3
1.1. Latar Belakang.....	3
1.2. Tujuan.....	3
1.3. Rumusan Masalah.....	3
1.4. Batasan Masalah.....	3
BAB II LANDASAN TEORI.....	4
2.1. K-Nearest Neighbor.....	4
2.2. Python.....	4
BAB III STUDI KASUS DAN IMPLEMENTASI.....	5
3.1. Studi Kasus.....	5
3.2. Implementasi Program.....	6
5) Mengklasifikasikan kelas berdasarkan data yang termasuk ke dalam tetangga dengan k = 3.....	11
BAB IV PENUTUP.....	12
4.1. Kesimpulan.....	12
DAFTAR PUSTAKA.....	13

BAB I PENDAHULUAN

1.1. Latar Belakang

Kesehatan tubuh manusia sangat dipengaruhi oleh banyak faktor, salah satunya adalah status gizi. Status gizi yang tidak seimbang dapat menyebabkan berbagai masalah kesehatan, baik dalam bentuk kekurangan gizi maupun obesitas. Oleh karena itu, identifikasi status gizi seseorang menjadi penting dalam upaya pencegahan dan penanganan masalah kesehatan.

Dalam dunia data mining, klasifikasi merupakan salah satu teknik yang dapat digunakan untuk mengidentifikasi status gizi berdasarkan berbagai parameter tubuh, seperti tinggi badan, berat badan, persentase lemak, tekanan darah, dan detak jantung. Klasifikasi memungkinkan kita untuk mengelompokkan data ke dalam kategori tertentu berdasarkan fitur-fitur yang tersedia.

Salah satu algoritma yang populer dalam klasifikasi adalah *K-Nearest Neighbors* (KNN). Algoritma ini bekerja dengan cara mencari data terdekat berdasarkan jarak dari data baru yang ingin diklasifikasikan. Pada penelitian ini, algoritma KNN diterapkan untuk mengklasifikasikan status gizi individu berdasarkan parameter fisik yang telah dikumpulkan, seperti tinggi badan, berat badan, persentase lemak, dan ukuran lingkaran tubuh.

1.2. Tujuan

1. Mengklasifikasikan status gizi individu berdasarkan parameter-parameter fisik seperti tinggi badan, berat badan, persentase lemak, tekanan darah, detak jantung, serta lingkaran tubuh menggunakan algoritma *K-Nearest Neighbors* (KNN).
2. Membangun model klasifikasi yang dapat membantu memprediksi status gizi seseorang (normal, kurus, obesitas) berdasarkan data yang tersedia.

1.3. Rumusan Masalah

1. Bagaimana algoritma *K-Nearest Neighbors* (KNN) dapat digunakan untuk mengklasifikasikan status gizi seseorang berdasarkan parameter fisik seperti tinggi badan, berat badan, persentase lemak, tekanan darah, detak jantung, serta ukuran lingkaran tubuh?
2. Seberapa akurat algoritma *K-Nearest Neighbors* (KNN) dalam mengidentifikasi status gizi seseorang?

1.4. Batasan Masalah

1. Akurasi klasifikasi dengan menggunakan algoritma KNN pada data ini hanya mencapai 60,00%, sehingga hasil prediksi yang diberikan oleh model mungkin masih kurang optimal. Hal ini menunjukkan adanya keterbatasan dalam memodelkan status gizi secara akurat berdasarkan parameter fisik yang digunakan.
2. Parameter yang digunakan dalam proses klasifikasi status gizi terbatas pada tinggi badan, berat badan, persentase lemak tubuh, tekanan darah (sistolik dan diastolik), detak jantung, ukuran lingkaran lengan (atas dan bawah), serta lingkaran perut dan pinggul. Variabel lain yang mungkin memengaruhi status gizi, seperti asupan makanan atau aktivitas fisik, tidak dimasukkan dalam model.

BAB II LANDASAN TEORI

2.1. K-Nearest Neighbor

K-Nearest Neighbor (KNN) merupakan metode yang biasa digunakan pada klasifikasi data. Algoritma ini digunakan untuk mengklasifikasikan terhadap objek berdasarkan data pembelajaran yang jarak tetangganya paling dekat atau memiliki nilai selisih yang kecil dengan objek tersebut. KNN merupakan suatu metode yang menggunakan algoritma supervised dengan hasil dari query instance yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada KNN. Tujuan dari algoritma ini ialah mengklasifikasikan objek baru berdasarkan attribut dan training sample. Prinsip umum dari algoritma ini adalah menemukan k data training untuk menentukan k-nearest neighbor berdasarkan ukuran jarak. Selanjutnya mayoritas dari k tetangga terdekat akan menjadi dasar untuk memutuskan kategori dari sample berikutnya. Selain itu algoritma ini sendiri sering digunakan untuk klasifikasi pada teknik data mining meskipun dapat digunakan untuk estimasi dan prediksi data.

2.2. Python

Python adalah bahasa pemrograman tingkat tinggi yang diciptakan oleh Guido van Rossum dan dirilis pertama kali pada tahun 1991. Dirancang dengan fokus pada kemudahan membaca dan menulis kode, Python menjadi pilihan utama di kalangan pengembang dan ilmuwan data. Sintaksis yang sederhana dan intuitif membuatnya mudah dipahami, baik oleh pemula maupun pengembang berpengalaman. salah satu kekuatan Python terletak pada ekosistem pustakanya yang luas. Pustaka-pustaka seperti NumPy untuk komputasi numerik, Pandas untuk manipulasi data, dan Scikit-learn untuk pembelajaran mesin memberikan alat yang kuat untuk analisis data dan pengembangan model prediktif. Dengan dukungan komunitas yang besar, pengguna dapat dengan mudah menemukan sumber daya dan bantuan.

Python juga mendukung pemrograman fungsional dan modular, memungkinkan pengembang untuk menulis kode yang terorganisir dan dapat digunakan kembali. Fleksibilitas ini menjadikannya ideal untuk berbagai aplikasi, termasuk pengembangan web, otomatisasi, dan analisis data. Dengan semua keunggulannya, Python telah menjadi bahasa utama dalam bidang data science dan pembelajaran mesin, memfasilitasi eksplorasi dan pengembangan model analitis yang efektif.

BAB III STUDI KASUS DAN IMPLEMENTASI

3.1. Studi Kasus

Studi kasus yang kami gunakan yaitu tentang mengklasifikasikan status gizi berdasarkan berbagai aspek. Aspek aspek yang ditinjau yaitu tinggi badan, berat badan, lemak (%), sistolik, distolik, detak jantung, lengan atas, lengan bawah, lingkar perut, lingkar pinggul. Untuk kategori dari status gizi terbagi menjadi tiga jenis yaitu kurus, normal, dan obesitas. Untuk data sampelnya dapat ditampilkan seperti ini:

Tinggi Badan	Berat Badan	Lemak (%)	Sistolik	Distolik	Detak Jantung	Lengan Atas	Lengan Bawah	Lingkar Perut	Lingkar Pinggul	Status Gizi
163	59	25.4	102	68	69	23	14	74	94	Normal
170	125	42.9	113	73	79	40	19	112	135	Obesitas
172	75	31	114	78	86	26	16	79	100	Normal
166	58	19	128	69	69	26	16	72	92	Normal
167	50	16.5	111	80	89	21	13	71	88	Kurus
168	50	10.4	100	72	97	23	14	62	87	Kurus
173	56	18.4	114	73	62	24	15	66	93	Normal
168	73	22.7	107	73	81	30	18	77	96	Normal
177	60	17.4	107	71	104	24	15	71	90	Normal
168	52	13.9	122	82	101	22	15	68	84	Kurus
159	58	23.4	118	75	64	26	15	70	85	Normal
167	75	30.7	126	81	65	30	16	91	103	Obesitas
170	72	26.1	113	72	83	28	16	85	98	Normal
172	68	22.8	110	65	68	27	15	79	94	Normal
165	73	29.1	105	67	75	28	18	83	100	Obesitas
169.5	55	15.2	112	77	70	22	14	75	92	Kurus
160	54	15.7	138	104	78	27	15	73	86	Normal
173	56	17.9	120	76	97	25	14	72	88	Normal
162	54	18	108	70	76	24	15	71	88	Normal
169	79	22	123	76	70	29	17	84	101	Obesitas

Untuk data baru yang akan kami klasifikasikan menggunakan algoritma k-nearest neighbor yaitu sebagai berikut:

Tinggi Badan (cm)	Berat Badan (kg)	Lemak (%)	Sistolik	Distolik	Detak Jantung (menit)	Lengan Atas (cm)	Lengan Bawah (cm)	Lingkar Perut (cm)	Lingkar Pinggul (cm)
175	67	23	97	67	79	25	15	75	95

3.2. Implementasi Program

1) Import library

```
1 # Importing necessary libraries
2 import pandas as pd
3 from sklearn.model_selection import train_test_split
4 from sklearn.neighbors import KNeighborsClassifier
5 from sklearn.preprocessing import LabelEncoder
6 from sklearn.impute import SimpleImputer
7 from sklearn.metrics import accuracy_score
8 import numpy as np
9 import math
10
```

2) Menampilkan data dari excel

```
1 # Path to the Excel file
2 file_path = r"data uji.xlsx"
3
4 # Read data from Excel
5 df = pd.read_excel(file_path, sheet_name="Sheet1")
6
7 # Convert necessary columns to numeric
8 kolom_numerik = [
9     "Tinggi Badan",
10    "Berat Badan",
11    "% Lemak",
12    "Sistolik",
13    "Diastolik",
14    "Detak Jantung",
15    "Lengan Atas",
16    "Lengan Bawah",
17    "Lingkar Perut",
18    "Lingkar Panggul",
19 ]
20 for kolom in kolom_numerik:
21     df[kolom] = pd.to_numeric(df[kolom], errors="coerce")
22
23 # Drop rows with NaN values
24 df = df.dropna()
25
26 # Tampilkan data dari Excel
27 print("### Data dari Excel:\n", df.to_string(), "\n")
28
```

Output:

### Data dari Excel:									
	Tinggi Badan	Berat Badan	% Lemak	Sistolik	Diastolik	Detak Jantung	Lengan Atas	Lengan	
Bawah	Lingkar Perut	Lingkar Panggul	Status Gizi						
0	163.0	59.0	25.4	102.0	68.0	69.0	23.0	14.0	74.0
94.0	Normal								
1	170.0	125.0	42.9	133.0	73.0	79.0	40.0	19.0	112.0
135.0	Obesitas								
2	172.0	75.0	31.0	114.0	79.0	86.0	26.0	16.0	79.0
100.0	Normal								
3	166.0	58.0	19.0	128.0	69.0	69.0	26.0	16.0	72.0
92.0	Normal								
4	167.0	50.0	16.5	111.0	80.0	89.0	21.0	13.0	71.0
88.0	Kurus								
5	168.0	50.0	10.4	100.0	72.0	97.0	23.0	14.0	62.0
87.0	Kurus								
6	173.0	56.0	18.4	114.0	73.0	62.0	24.0	15.0	66.0
93.0	Normal								
7	168.0	73.0	22.7	107.0	73.0	81.0	30.0	18.0	77.0
96.0	Normal								
8	177.0	60.0	17.4	107.0	71.0	104.0	24.0	15.0	71.0
90.0	Normal								
9	168.0	52.0	13.9	122.0	81.0	101.0	22.0	15.0	68.0
84.0	Kurus								
10	159.0	58.0	23.4	118.0	75.0	64.0	26.0	15.0	70.0
85.0	Normal								
11	167.0	75.0	30.7	126.0	81.0	65.0	30.0	16.0	91.0
103.0	Obesitas								
12	170.0	72.0	26.1	113.0	72.0	83.0	28.0	16.0	85.0
98.0	Normal								
13	172.0	68.0	22.8	110.0	65.0	68.0	27.0	15.0	79.0
94.0	Normal								
14	165.0	73.0	29.1	105.0	67.0	75.0	28.0	18.0	83.0
100.0	Obesitas								
15	169.5	55.0	15.2	112.0	77.0	70.0	22.0	14.0	75.0
92.0	Kurus								
16	160.0	54.0	15.7	138.0	104.0	78.0	27.0	15.0	73.0
86.0	Normal								
17	173.0	56.0	17.9	120.0	6.0	97.0	25.0	14.0	72.0
88.0	Normal								
18	162.0	54.0	18.0	108.0	70.0	76.0	14.0	15.0	71.0
88.0	Normal								
19	169.0	79.0	22.0	123.0	76.0	70.0	29.0	17.0	84.0
101.0	Obesitas								
24	175.0	67.0	23.0	97.0	67.0	79.0	25.0	15.0	75.0
96.0	?								

3) Menghitung jarak euclidian

```
1 # Step 3: Preprocess the data
2 le = LabelEncoder()
3 df["Status Gizi Terencoded"] = le.fit_transform(df["Status Gizi"])
4
5 # Features (X) and target (y)
6 X = df.drop(columns=["Status Gizi", "Status Gizi Terencoded"])
7 y = df["Status Gizi Terencoded"]
8
9 # Define the new data point (referensi)
10 referensi = {
11     "Tinggi Badan": 175,
12     "Berat Badan": 67,
13     "% Lemak": 23,
14     "Sistolik": 97,
15     "Diastolik": 67,
16     "Detak Jantung": 79,
17     "Lengan Atas": 25,
18     "Lengan Bawah": 15,
19     "Lingkar Perut": 75,
20     "Lingkar Panggul": 95,
21 }
22
23 # Fungsi untuk menghitung Euclidean Distance
24 def jarak_euclidean(baris, ref):
25     jarak = 0
26     for kolom in ref.keys():
27         jarak += (baris[kolom] - ref[kolom])**2
28     return math.sqrt(jarak)
29
30 # Menghitung Jarak (Euclidean Distance) untuk 20 data pertama
31 df["Jarak Euclidean"] = df.iloc[:20].apply(jarak_euclidean, ref=referensi, axis=1)
32
33 # Menampilkan hasil perhitungan jarak
34 print("\n### Hasil Perhitungan Jarak (Euclidean Distance)")
35 for i, baris in df.iloc[:20].iterrows():
36     print(
37         f"Data {i+1}\n"
38         f"dis = sqrt((Tinggi Badan - {referensi['Tinggi Badan']})^2 + "
39         f"{(Berat Badan - {referensi['Berat Badan']})^2 + "
40         f"{(Lemak - {referensi['% Lemak']})^2 + "
41         f"{(Sistolik - {referensi['Sistolik']})^2 + "
42         f"{(Diastolik - {referensi['Diastolik']})^2 + "
43         f"{(Detak Jantung - {referensi['Detak Jantung']})^2 + "
44         f"{(Lengan Atas - {referensi['Lengan Atas']})^2 + "
45         f"{(Lengan Bawah - {referensi['Lengan Bawah']})^2 + "
46         f"{(Lingkar Perut - {referensi['Lingkar Perut']})^2 + "
47         f"{(Lingkar Panggul - {referensi['Lingkar Panggul']})^2) = {baris['Jarak Euclidean']:.2f}\n"
48     )
```

Output:

Hasil Perhitungan Jarak (Euclidean Distance)

Data 1

$$\text{dis} = \sqrt{(\text{Tinggi Badan} - 175)^2 + (\text{Berat Badan} - 67)^2 + (\% \text{ Lemak} - 23)^2 + (\text{Sistolik} - 97)^2 + (\text{Diastolik} - 67)^2 + (\text{Detak Jantung} - 79)^2 + (\text{Lengan Atas} - 25)^2 + (\text{Lengan Bawah} - 15)^2 + (\text{Lingkar Perut} - 75)^2 + (\text{Lingkar Panggul} - 95)^2} = 18.62$$

Data 2

$$\text{dis} = \sqrt{(\text{Tinggi Badan} - 175)^2 + (\text{Berat Badan} - 67)^2 + (\% \text{ Lemak} - 23)^2 + (\text{Sistolik} - 97)^2 + (\text{Diastolik} - 67)^2 + (\text{Detak Jantung} - 79)^2 + (\text{Lengan Atas} - 25)^2 + (\text{Lengan Bawah} - 15)^2 + (\text{Lingkar Perut} - 75)^2 + (\text{Lingkar Panggul} - 95)^2} = 91.25$$

Data 3

$$\text{dis} = \sqrt{(\text{Tinggi Badan} - 175)^2 + (\text{Berat Badan} - 67)^2 + (\% \text{ Lemak} - 23)^2 + (\text{Sistolik} - 97)^2 + (\text{Diastolik} - 67)^2 + (\text{Detak Jantung} - 79)^2 + (\text{Lengan Atas} - 25)^2 + (\text{Lengan Bawah} - 15)^2 + (\text{Lingkar Perut} - 75)^2 + (\text{Lingkar Panggul} - 95)^2} = 25.73$$

Data 4

$$\text{dis} = \sqrt{(\text{Tinggi Badan} - 175)^2 + (\text{Berat Badan} - 67)^2 + (\% \text{ Lemak} - 23)^2 + (\text{Sistolik} - 97)^2 + (\text{Diastolik} - 67)^2 + (\text{Detak Jantung} - 79)^2 + (\text{Lengan Atas} - 25)^2 + (\text{Lengan Bawah} - 15)^2 + (\text{Lingkar Perut} - 75)^2 + (\text{Lingkar Panggul} - 95)^2} = 35.54$$

$$(\text{Lingkar Perut} - 75)^2 + (\text{Lingkar Panggul} - 95)^2 = 18.98$$

Data 16

$$\text{dis} = \sqrt{(\text{Tinggi Badan} - 175)^2 + (\text{Berat Badan} - 67)^2 + (\% \text{ Lemak} - 23)^2 + (\text{Sistolik} - 97)^2 + (\text{Diastolik} - 67)^2 + (\text{Detak Jantung} - 79)^2 + (\text{Lengan Atas} - 25)^2 + (\text{Lengan Bawah} - 15)^2 + (\text{Lingkar Perut} - 75)^2 + (\text{Lingkar Panggul} - 95)^2} = 25.69$$

Data 17

$$\text{dis} = \sqrt{(\text{Tinggi Badan} - 175)^2 + (\text{Berat Badan} - 67)^2 + (\% \text{ Lemak} - 23)^2 + (\text{Sistolik} - 97)^2 + (\text{Diastolik} - 67)^2 + (\text{Detak Jantung} - 79)^2 + (\text{Lengan Atas} - 25)^2 + (\text{Lengan Bawah} - 15)^2 + (\text{Lingkar Perut} - 75)^2 + (\text{Lingkar Panggul} - 95)^2} = 59.89$$

Data 18

$$\text{dis} = \sqrt{(\text{Tinggi Badan} - 175)^2 + (\text{Berat Badan} - 67)^2 + (\% \text{ Lemak} - 23)^2 + (\text{Sistolik} - 97)^2 + (\text{Diastolik} - 67)^2 + (\text{Detak Jantung} - 79)^2 + (\text{Lengan Atas} - 25)^2 + (\text{Lengan Bawah} - 15)^2 + (\text{Lingkar Perut} - 75)^2 + (\text{Lingkar Panggul} - 95)^2} = 69.17$$

Data 19

$$\text{dis} = \sqrt{(\text{Tinggi Badan} - 175)^2 + (\text{Berat Badan} - 67)^2 + (\% \text{ Lemak} - 23)^2 + (\text{Sistolik} - 97)^2 + (\text{Diastolik} - 67)^2 + (\text{Detak Jantung} - 79)^2 + (\text{Lengan Atas} - 25)^2 + (\text{Lengan Bawah} - 15)^2 + (\text{Lingkar Perut} - 75)^2 + (\text{Lingkar Panggul} - 95)^2} = 26.23$$

Data 20

$$\text{dis} = \sqrt{(\text{Tinggi Badan} - 175)^2 + (\text{Berat Badan} - 67)^2 + (\% \text{ Lemak} - 23)^2 + (\text{Sistolik} - 97)^2 + (\text{Diastolik} - 67)^2 + (\text{Detak Jantung} - 79)^2 + (\text{Lengan Atas} - 25)^2 + (\text{Lengan Bawah} - 15)^2 + (\text{Lingkar Perut} - 75)^2 + (\text{Lingkar Panggul} - 95)^2} = 34.00$$

4) Mengurutkan data berdasarkan jarak euclidian

```
1 # Mengurutkan data berdasarkan Jarak Euclidean dari terkecil ke terbesar
2 df_sorted = df.iloc[:20].sort_values(by="Jarak Euclidean")
3
4 # Menampilkan data yang telah diurutkan
5 print("\n### Data yang Diurutkan Berdasarkan Jarak Euclidean (Terkecil ke Terbesar):")
6 print(df_sorted[["Jarak Euclidean", "Status Gizi"]])
7
```

Output:

Data yang Diurutkan Berdasarkan Jarak Euclidean (Terkecil ke Terbesar):

	Jarak Euclidean	Status Gizi
7	16.250846	Normal
13	18.028866	Normal
0	18.621493	Normal
14	18.979199	Obesitas
12	21.808485	Normal
15	25.692217	Kurus

2	25.729361	Normal
18	26.229754	Normal
6	29.088829	Normal
8	29.450976	Normal
4	30.744918	Kurus
5	33.056921	Kurus
19	34.000000	Obesitas
10	34.542148	Normal
3	35.538711	Normal
11	42.027253	Obesitas
9	42.904662	Kurus
16	59.893990	Normal
17	69.166538	Normal
1	91.252452	Obesitas

- 5) Mengklasifikasikan kelas berdasarkan data yang termasuk ke dalam tetangga dengan $k = 3$

```

1 # Impute missing values
2 imputer = SimpleImputer(strategy='mean')
3 X_imputed = imputer.fit_transform(X)
4
5 # Train-test split with imputed data
6 X_latih, X_uji, y_latih, y_uji = train_test_split(X_imputed, y, test_size=0.2, random_state=42)
7
8 # Step 8: KNN model training
9 knn = KNeighborsClassifier(n_neighbors=3)
10 knn.fit(X_latih, y_latih)
11
12 # Step 9: Predict the new data point
13 prediksi = knn.predict(
14     [list(referensi.values())])
15 ) # Ubah referensi ke list untuk prediksi
16 prediksi_label = le.inverse_transform(prediksi)
17 print(f'\nStatus Gizi yang diprediksi untuk data baru adalah: {prediksi_label[0]}")
18
19 # Step 10: Evaluate the model (optional)
20 y_pred = knn.predict(X_uji)
21
22 # Calculate accuracy
23 y_pred = knn.predict(X_uji)
24 akurasi = accuracy_score(y_uji, y_pred)
25 print(f"Akurasi model: {akurasi * 100:.2f}%")
26

```

Output:

Status Gizi yang diprediksi untuk data baru adalah: Normal
 Akurasi model: 60.00%

BAB IV PENUTUP

4.1. Kesimpulan

Dalam studi kasus ini menggunakan algoritma K-Nearest Neighbors (KNN) untuk mengklasifikasikan status gizi berdasarkan parameter fisik seperti tinggi badan, berat badan, persentase lemak tubuh, tekanan darah, detak jantung, serta ukuran lingkar tubuh. Berdasarkan hasil analisis, model KNN yang dibangun hanya mencapai tingkat akurasi sebesar 60,00%. Hasil akurasi ini menunjukkan bahwa model belum optimal dalam mengidentifikasi status gizi. Akurasi yang rendah ini kemungkinan disebabkan oleh keterbatasan pada variabel yang digunakan serta jumlah dan keragaman data.

DAFTAR PUSTAKA

<https://jurnal.mdp.ac.id/index.php/jatisi/article/download/2048/785>

<http://repository.uin-suska.ac.id/3012/3/BAB%20II.pdf>