

BAB 3

METODE PENELITIAN

Penelitian ini menggunakan pendekatan analisis faktor-faktor yang memengaruhi stunting pada anak balita di Kabupaten Klaten. Data akan dikumpulkan melalui survei lapangan menggunakan wawancara dengan pihak dinas kesehatan Kabupaten Klaten yang mencakup karakteristik demografis, status gizi, akses terhadap layanan kesehatan, sanitasi lingkungan. Selanjutnya, algoritma KNN akan diterapkan untuk mengidentifikasi faktor-faktor yang paling berpengaruh terhadap *stunting*.

3.1 BAHAN DAN ALAT PENELITIAN

Bahan penelitian yang diperlukan meliputi data primer dan sekunder. Data primer mencakup informasi seperti tinggi badan, berat badan, lingkar kepala, status gizi, dan riwayat kesehatan anak balita di Kabupaten Klaten. Selain itu, data mengenai faktor internal seperti faktor genetik, kelainan bawaan, prematuritas, dan berat badan lahir rendah juga diperlukan untuk analisis. Data sekunder, yang berasal dari penelitian sebelumnya mengenai stunting di Kabupaten Klaten serta informasi tentang praktik kesehatan masyarakat lokal, dapat memberikan tambahan wawasan. Selanjutnya, pemahaman tentang algoritma KNN dari literatur ilmiah perlu diperoleh untuk memahami metodologi analisis yang digunakan dalam penelitian ini.

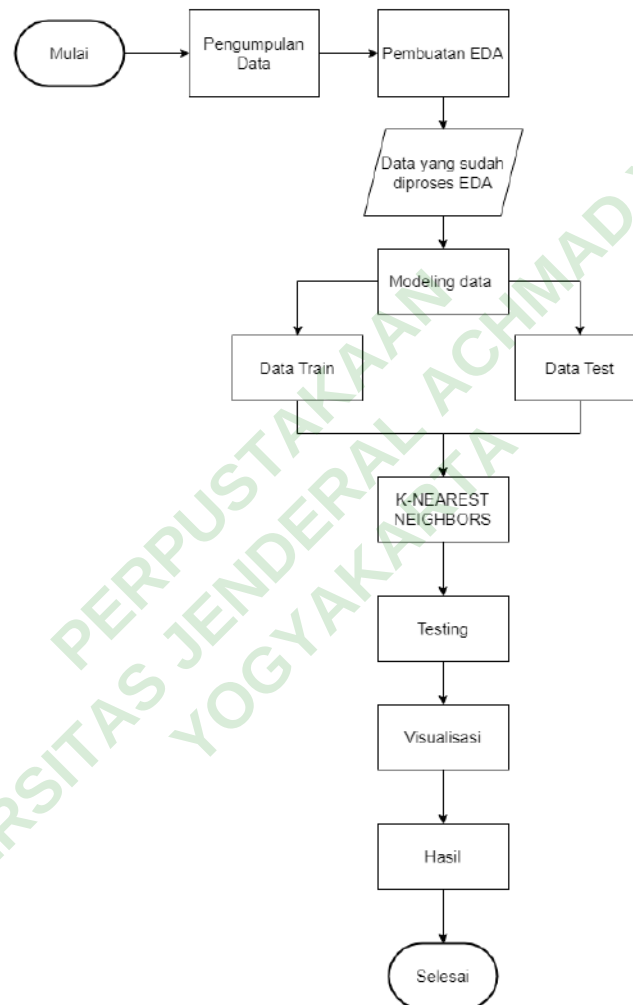
Alat yang digunakan dalam penelitian ini adalah komputer dengan spesifikasi yang memadai untuk menjalankan sistem operasi dan perangkat lunak pengembangan, dan tersambung ke Internet.

Sistem operasi dan aplikasi yang diterapkan dalam pengembangan aplikasi ini adalah:

1. Sistem Operasi : Windows 10
2. Bahasa Pemrograman : Python dan *Framework* Flask
3. Aplikasi Tambahan : Visual Studio Code, *Jupyter* Notebook.

3.2 JALAN PENELITIAN

Pada tahapan ini akan dilakukan secara berurutan sesuai dengan alur penelitian. Alur penelitian serta tahapan KNN yang akan digunakan dalam penelitian tugas akhir, ditunjukkan pada Gambar 3.1.



Gambar 3.1 Alur penelitian

3.2.1 Pengumpulan Data

Langkah pertama adalah mengumpulkan *dataset* yang relevan untuk penelitian, yakni *dataset* yang berhubungan dengan informasi kesehatan dan mencakup variabel-variabel penting untuk analisis faktor-faktor yang memengaruhi *stunting*.

[illegible]

Selanjutnya data rekapan tersebut dipecah dan mendapatkan data terdiri dari 55160 data dan dilakukan pelabelan hasil *stunting* berupa positif dan negatif secara manual dengan melakukan pertimbangan dan sesuai dengan peraturan permenkes (Kementerian Kesehatan, 2020). Data yang telah dipecah, dapat dilihat pada Gambar 3.3.

	Puskesmas	Umur (bulan)	Jenis kelamin	sangat kur	kur	bb normal	risiko leti	gizi buruk	gizi kurang	gizi normal	gizi let	gizi lebih	obesitas	ang	pent	pendek	tb normal	tb tinggi	stunting
1	PRAMBAN	0-23 bln	perempuan	ya	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	positif
2	PRAMBAN	0-23 bln	laki-laki	ya	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	positif
3	PRAMBAN	0-23 bln	laki-laki	ya	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	positif
4	PRAMBAN	0-23 bln	laki-laki	ya	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	positif
5	PRAMBAN	0-23 bln	laki-laki	ya	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	positif
6	PRAMBAN	24-59 bln	laki-laki	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	positif
7	PRAMBAN	24-59 bln	laki-laki	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	positif
8	PRAMBAN	24-59 bln	laki-laki	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	positif
9	PRAMBAN	24-59 bln	laki-laki	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	positif
10	PRAMBAN	24-59 bln	laki-laki	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	positif
11	PRAMBAN	24-59 bln	perempuan	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	positif
12	PRAMBAN	24-59 bln	perempuan	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	positif
13	PRAMBAN	24-59 bln	perempuan	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	positif
14	PRAMBAN	24-59 bln	perempuan	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	positif
15	PRAMBAN	24-59 bln	perempuan	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	positif
16	PRAMBAN	0-23 bln	perempuan	tidak	ya	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	positif
17	PRAMBAN	0-23 bln	perempuan	tidak	ya	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	positif
18	PRAMBAN	0-23 bln	perempuan	tidak	ya	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	positif
19	PRAMBAN	0-23 bln	perempuan	tidak	ya	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	positif
20	PRAMBAN	0-23 bln	perempuan	tidak	ya	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	positif
21	PRAMBAN	0-23 bln	perempuan	tidak	ya	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	positif
22	PRAMBAN	0-23 bln	perempuan	tidak	ya	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	positif

Data yang telah dipecah akan digunakan dalam melakukan penelitaian dengan atribut yang sesuai di dalam data. Atribut-atribut yang digunakan dalam *dataset* :

Tabel 3.1 Atribut yang digunakan dalam *dataset*

No	Nama Atribut	Keterangan
1	Puskesmas	Nama Puskesmas
2	Umur	0-59 Bulan
3	Jenis Kelamin	Laki-laki/Perempuan
4	Berat badan sangat kurang	Ya/Tidak
5	Berat badan kurang	Ya/Tidak
6	Berat badan normal	Ya/Tidak
7	Berat badan risiko lebih	Ya/Tidak
8	Gizi buruk	Ya/Tidak
9	Gizi kurang	Ya/Tidak
10	Gizi normal	Ya/Tidak
11	Risiko gizi lebih	Ya/Tidak
12	Gizi lebih	Ya/Tidak
13	Obesitas	Ya/Tidak
14	Tinggi badan sangat pendek	Ya/Tidak
15	Tinggi badan pendek	Ya/Tidak
16	Tinggi badan normal	Ya/Tidak
17	Tinggi badan tinggi	Ya/Tidak
18	<i>Stunting</i>	Positif/Negatif

3.2.2 EDA (Exploratory Data Analysis)

Eksplorasi Data Awal (EDA) adalah proses krusial dalam analisis data yang membantu memahami struktur, pola, dan anomali dalam dataset. Langkah-langkah ini mencakup pemeriksaan ukuran data, identifikasi nilai yang hilang, penghapusan baris kosong, verifikasi ulang nilai yang hilang, pengecekan tipe data, konversi data kategori menjadi numerik, dan pengecekan ulang tipe data. Berikut adalah langkah-langkah untuk melakukan EDA:

1. Memeriksa Ukuran Data

Langkah yang pertama dalam EDA adalah memeriksa jumlah kolom dan baris dalam *dataset* untuk memahami ukuran data yang akan dianalisis, seperti pada kode dibawah ini.

```
# memeriksa ukuran data
df.shape
```

Output yang diperoleh dari menjalankan kode tersebut terlihat pada Gambar 3.4 dengan jumlah data yang digunakan berjumlah 55160 data dan panjang baris kolom berjumlah 18 data.

```
Out[3]: (55160, 18)
```

Gambar 3.4 *Output* kode memeriksa ukuran data

2. Memeriksa Nilai Yang Hilang

Langkah berikutnya dalam tahap EDA adalah memeriksa keberadaan nilai yang hilang (*missing values*) dalam *dataset* dapat berdampak pada analisis dan hasil model. Seperti pada kode dibawah ini.

```
# Memeriksa nilai yang hilang
df.isnull().sum()
```

Output kode tersebut terdapat 8 kolom nilai hilangnya sama yaitu kolom gizi buruk, gizi kurang, gizi normal, risiko gizi lebih, gizi lebih, obesitas, tb sangat pendek, tb pendek berjumlah 489 nilai yang hilang, sedangkan tb normal berjumlah 487 nilai yang hilang dan tb tinggi berjumlah 493 nilai yang hilang, dapat dilihat pada Gambar 3.5.

```
Out[4]: Puskesmas          0
        Umur (bulan)        0
        Jenis kelamin       0
        bb sangat kurang    0
        bb kurang           0
        bb normal           0
        bb risiko lebih     0
        gizi buruk          489
        gizi kurang         489
        gizi normal         489
        risiko gizi lebih   489
        gizi lebih          489
        obesitas            489
        tb sangat pendek    489
        tb pendek           489
        tb normal           487
        tb tinggi           493
        stunting            0
        dtype: int64
```

Gambar 3.5 *Output* kode memeriksa nilai yang hilang

3. Menghapus Baris Yang Kosong

Langkah berikutnya adalah menghapus baris yang kosong untuk memudahkan dalam proses EDA. Baris kosong dapat memengaruhi analisis dan hasil model, sehingga perlu di tanganni terlebih dahulu. Berikut kode untuk menghapus baris yang kosong, dapat dilihat seperti pada kode dibawah.

```
# menghapus baris yang kosong
df.dropna(inplace=True)
df
```

Kode tersebut digunakan untuk menghapus baris yang kosong dalam *dataset*. *Output* menunjukan baris yang kosong telah ditangani pada dalam *dataset*. Pada hasil tersebut berjumlah 54605 kolom dan 18 baris.

out[38]:

	Puskesmas	Umur (bulan)	Jenis kelamin	bb sangat kurang	bb kurang	bb normal	bb risiko lebih	gizi buruk	gizi kurang	gizi normal	risiko gizi lebih	gizi lebih	obesitas	tb sangat pendek	tb pendek	tb normal	tb tinggi	stun
0	PRAMBANAN	0-23 bin	perempuan	ya	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	pc
1	PRAMBANAN	0-23 bin	laki-laki	ya	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	pc
2	PRAMBANAN	0-23 bin	laki-laki	ya	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	pc
3	PRAMBANAN	0-23 bin	laki-laki	ya	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	pc
4	PRAMBANAN	24-59 bin	laki-laki	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	pc
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
55116	KLATEN UTARA	24-59 bin	laki-laki	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	ya	nej
55117	KLATEN UTARA	24-59 bin	laki-laki	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	ya	nej
55118	KLATEN UTARA	24-59 bin	laki-laki	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	ya	nej
55119	KLATEN UTARA	24-59 bin	laki-laki	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	ya	nej
55120	KLATEN UTARA	24-59 bin	laki-laki	tidak	tidak	tidak	ya	tidak	tidak	tidak	tidak	tidak	ya	tidak	tidak	tidak	ya	nej

54605 rows x 18 columns

Gambar 3.6 Output kode menghapus data yang kosong

4. Mengecek Kembali Nilai Yang Hilang

Langkah berikutnya adalah memeriksa kembali apakah masih ada nilai yang hilang dalam *dataset* setelah menghapus baris kosong. Kode yang digunakan seperti kode dibawah.

```
# mengecek kembali nilai yang hilang
df.isna().sum()
```

Output dari kode menunjukkan tidak ada kolom yang memiliki nilai kosong pada *dataset*, dapat dilihat pada Gambar 3.7.

```
Out[6]: Puskesmas      0
        Umur (bulan)    0
        Jenis kelamin   0
        bb sangat kurang 0
        bb kurang       0
        bb normal       0
        bb risiko lebih 0
        gizi buruk      0
        gizi kurang     0
        gizi normal     0
        risiko gizi lebih 0
        gizi lebih      0
        obesitas        0
        tb sangat pendek 0
        tb pendek       0
        tb normal       0
        tb tinggi       0
        stunting        0
        dtype: int64
```

Gambar 3.7 *Output* kode mengecek kembali nilai yang hilang

5. Mengecek Tipe Data

Langkah berikutnya adalah melakukan pemeriksaan terhadap tipe data setiap kolom untuk memverifikasi kesesuaian dengan jenis data yang diharapkan (seperti numerik atau teks), seperti pada kode dibawah ini.

```
# cek tipe data
df.info()
```

Hasil menunjukkan tipe data dari setiap kolom. Kolom pada *dataset* tersebut bertipe *object* (teks), dapat dilihat pada Gambar 3.8.

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 54605 entries, 0 to 55120
Data columns (total 18 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Puskesmas             54605 non-null  object
1   Umur (bulan)          54605 non-null  object
2   Jenis kelamin         54605 non-null  object
3   bb sangat kurang      54605 non-null  object
4   bb kurang             54605 non-null  object
5   bb normal             54605 non-null  object
6   bb risiko lebih      54605 non-null  object
7   gizi buruk            54605 non-null  object
8   gizi kurang           54605 non-null  object
9   gizi normal           54605 non-null  object
10  risiko gizi lebih     54605 non-null  object
11  gizi lebih            54605 non-null  object
12  obesitas              54605 non-null  object
13  tb sangat pendek      54605 non-null  object
14  tb pendek             54605 non-null  object
15  tb normal             54605 non-null  object
16  tb tinggi             54605 non-null  object
17  stunting              54605 non-null  object
dtypes: object(18)

```

Gambar 3.8 Output kode mengecek tipe data

6. Konversi Data Kategori Ke Numerik

Kolom yang mengandung data kategori dikonversi menjadi data numerik digunakan untuk analisis. Konversi yang dilakukan dari bentuk teks menjadi angka, seperti teks “tidak” akan berubah menjadi angka “0” dan teks “ya” akan berubah menjadi angka “1”. Pemecahan kolom ini dapat dilakukan dengan menggunakan Label_Encoder untuk mengubah data kategori menjadi angka. Kode yang digunakan, seperti kode dibawah ini.

```

# konversi data kategori ke numerik
label_encoder = LabelEncoder()
for column in df.columns:
    df[column] = label_encoder.fit_transform(df[column])
df.head()

```

Output pada kode telah dirubah dari kategori ke numerik, dapat dilihat pada Gambar 3.9.

	Puskesmas	Umur (bulan)	Jenis kelamin	bb sangat kurang	bb kurang	bb normal	bb risiko lebih	gizi buruk	gizi kurang	gizi normal	risiko gizi lebih	gizi lebih	obesitas	tb sangat pendek	tb pendek	tb normal	tb tinggi	stunting
0	27	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0	0	1
1	27	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	1
2	27	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	1
3	27	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	1
4	27	2	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	1

Gambar 3.9 Output kode konversi data kategori ke numerik

7. Menampilkan Informasi Setelah Konversi Data Kategori ke Numerik

Selanjutnya menampilkan informasi data kembali untuk memastikan data yang semula dalam bentuk kategori telah berhasil dikonversi menjadi data numerik. Informasi ini penting sebelum melanjutkan ke tahap pemodelan data. Kode yang digunakan untuk menampilkan data setelah dikonversi, seperti pada kode dibawah.

```
# mengecek tipe data kembali
df.info()
```

Output menunjukkan bahwa tidak ada nilai yang hilang setelah konversi data ke numerik, dan tipe data sesuai. Pada hasil tersebut juga menunjukkan bahwa kolom yang sebelumnya bertipe kategori telah dikonversi menjadi numerik, dapat dilihat pada Gambar 3.10.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 54605 entries, 0 to 55120
Data columns (total 18 columns):
#   Column                Non-Null Count  Dtype
---  ---                ---
0   Puskesmas             54605 non-null  int32
1   Umur (bulan)          54605 non-null  int32
2   Jenis kelamin         54605 non-null  int32
3   bb sangat kurang      54605 non-null  int32
4   bb kurang             54605 non-null  int32
5   bb normal             54605 non-null  int32
6   bb risiko lebih      54605 non-null  int32
7   gizi buruk            54605 non-null  int32
8   gizi kurang           54605 non-null  int32
9   gizi normal           54605 non-null  int32
10  risiko gizi lebih     54605 non-null  int32
11  gizi lebih            54605 non-null  int32
12  obesitas              54605 non-null  int32
13  tb sangat pendek      54605 non-null  int32
14  tb pendek             54605 non-null  int32
15  tb normal             54605 non-null  int32
16  tb tinggi             54605 non-null  int32
17  stunting              54605 non-null  int32
dtypes: int32(18)
memory usage: 4.2 MB
```

Gambar 3.10 Output kode informasi data kategori ke numerik

3.2.3 Modeling Data

Modeling data merupakan tahap krusial dalam analisis data untuk memahami dan mempersiapkan data sebelum membangun model. Berikut langkah-langkah dalam modeling data.

1. Memisahkan Kumpulan Data Menjadi Fitur dan Variable Target

Tahapan awal dalam proses pemilihan fitur adalah memisahkan dataset menjadi dua bagian, yaitu fitur-fitur (X) dan variabel target (y). Fitur-fitur merupakan variabel yang berfungsi sebagai masukan untuk model, sedangkan variabel target adalah yang ingin diprediksi. Kode yang digunakan memisahkan fitur dan variable, seperti kode dibawah.

```
# Memisahkan kumpulan data menjadi fitur dan variabel target
X = df.drop(['Puskesmas', 'stunting'], axis=1)
y = df['stunting']
```

Dataset dibagi menjadi fitur dan variabel target. X mencakup fitur-fitur, sedangkan y mencakup variabel target. Fitur yang akan digunakan sebagai input dalam model analisis yaitu pada X kecuali kolom ['puskesmas', 'stunting'] sedangkan pada y nilai yang ingin diprediksi atau dianalisis adalah kolom label 'stunting'.

2. Memisahkan Data Menjadi Data Latih dan Data Uji

Langkah berikutnya adalah membagi data latih (*train*) dan data uji (*test*). Data latih digunakan untuk melatih model, sedangkan data uji digunakan untuk menguji performa model. Kode yang digunakan seperti pada kode dibawah.

```
# Memisahkan data menjadi data latih dan data uji
X_train, X_test, y_train, y_test = train_test_split
(X, y, test_size=0.2, stratify=y, random_state=42)
```

Kode tersebut memanfaatkan fungsi `train_test_split` dari pustaka `scikit-learn` dalam modul '`model_selection`' digunakan untuk memisahkan dataset menjadi dua bagian, yakni data pelatihan dan data pengujian. Pada tahap ini, X merujuk kepada himpunan fitur, sedangkan y merupakan variabel target. Dan kode `test_size=0.2` menentukan persentase data yang akan digunakan sebagai data uji dalam kode tersebut 20% dari data akan digunakan untuk pengujian, rasio 80:20 membantu

memastikan model dapat diuji pada data representatif dari populasi yang lebih luas dan memberikan evaluasi lebih akurat terhadap performa model (Bonardo et al., 2021). Kode `stratify=y` berfungsi sebagai memastikan proporsi kelas data latih dan data uji akan sama dengan proporsi kelas di *dataset*. Sedangkan `random_state=42` berfungsi untuk mengatur seed pada generator angka acak untuk memastikan kode yang dijalankan dengan nilai 'random_state' sama setiap kode dijalankan.

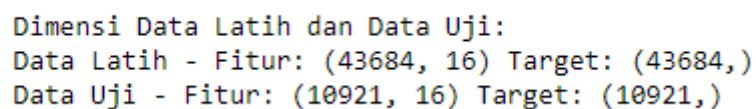
3. Memeriksa Dimensi Data Latih dan Data Uji

Langkah berikutnya adalah mencetak dimensi data latih (*train*) dan data uji (*test*) setelah membagi *dataset* menggunakan 'train_test_split'. Kode dapat dilihat pada kode dibawah ini.

```
# Memeriksa dimensi data latih dan data uji
print('Dimensi Data Latih dan Data Uji:')
print("Data Latih - Fitur:", X_train.shape, "Target:",
      y_train.shape)
print("Data Uji - Fitur:", X_test.shape, "Target:", y_test.shape)
```

`X_train.shape` berfungsi mengembalikan dimensi dari 'x_train' pada data fitur yang digunakan untuk melatih model. Sedangkan `y_train_shape` digunakan mengembalikan dimensi dari 'y_train' pada target yang sesuai dengan data fitur latih. Kode `x_test.shape` mengembalikan dimensi dari 'x_test' data fitur yang digunakan untuk menguji model dan `y_test.shape` mengembalikan dimensi dari 'y_test' target yang sesuai dengan data fitur uji.

Pada kode tersebut memperoleh hasil untuk data latih yaitu 43684 baris dan 16 fitur dan target untuk data latih sebesar 43684, sedangkan data uji yaitu 10921 baris dan 16 fitur dan target untuk data uji sebesar 10921. Sesuai dengan pembagian proporsi 80:20 yang ditentukan. Dapat dilihat pada Gambar 3.11.



```
Dimensi Data Latih dan Data Uji:
Data Latih - Fitur: (43684, 16) Target: (43684,)
Data Uji - Fitur: (10921, 16) Target: (10921,)
```

Gambar 3.11 Output kode dimensi data latih dan uji

3.2.4 Pembuatan Model

Pembuatan model merupakan tahap krusial dalam analisis, di mana algoritma digunakan untuk melakukan klasifikasi berdasarkan informasi yang tersedia. Pada tahap ini, digunakan algoritma KNN untuk membangun model klasifikasi analisis *stunting*. Langkah pertama adalah membuat model menggunakan algoritma KNN dengan kriteria data yang tersedia. Kode dapat dilihat pada kode dibawah.

```
# pembuatan model
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
```

Pada kode tersebut dijelaskan, `KNeighborsClassifier` dari `scikit-learn` digunakan untuk membuat model KNN dengan memberikan argument `n_neighbors=5` untuk menggunakan 5 tetangga terdekat untuk melakukan prediksi berdasarkan jarak *Euclidean*. Selanjutnya adalah melatih model menggunakan data latih dengan kode: `knn.fit (X_train, y_train)` ‘metode fit’ akan membuat model yang mempelajari hubungan antara fitur ‘`x_train`’ dan target ‘`y_train`’ sehingga model dapat digunakan untuk membuat prediksi pada data baru. *Output* pembuatan model knn dapat dilihat pada Gambar 3.12.

```
Out[13]: KNeighborsClassifier()
```

Gambar 3.12 *Output* pembuatan model