

LAMPIRAN

Lampiran 1 Tampilan Dataset Stunting Kota Samarinda

No	Nama	JK	BB Lahir	TB Lahir	Prov	Kab/Kota	Kec	Pukesmas	Desa/Kel	Posyandu	RT	RW	Tanggal Pengukuran	Berat	Tinggi	LILA	BB/U	ZS BB/U	TB/U	ZS TB/U	BB/TB	ZS BB/TB	Naik Berat Badan	Jml Vit A
1.	DIMAS ADITYA	L	3.5	49	KALIMANTAN TIMUR	SAMARINDA	SUNGAI PINANG	REMAJA	TEMINDUNG PERMAI	PULAU INDAH			2023-01-02	9.1	74.5	0	Berat Badan Normal	-0.39	Normal	-0.21	Gizi Baik	-0.39	O	
2.	SITI AISYAH	P	3	48	KALIMANTAN TIMUR	SAMARINDA	SUNGAI PINANG	REMAJA	TEMINDUNG PERMAI	PULAU INDAH	34		2023-01-02	12	94	0	Kurang	-2.25	Pendek	-2.09	Gizi Baik	-1.46	O	
3.	M AL FATHI	L	2.8	49	KALIMANTAN TIMUR	SAMARINDA	SUNGAI PINANG	REMAJA	TEMINDUNG PERMAI	PULAU INDAH			2023-01-02	8.1	69	0	Berat Badan Normal	-0.53	Normal	-0.65	Gizi Baik	-0.14	O	
4.	ALMAHIRA AKIRA AKBAR	P	3.98	50647201	KALIMANTAN TIMUR	SAMARINDA	SUNGAI PINANG	REMAJA	TEMINDUNG PERMAI	PULAU INDAH			2023-01-02	6.3	63.5	0	Berat Badan Normal	-0.31	Normal	0.42	Gizi Baik	-0.74	O	
5.	GIUNIA QAMELA	P	3.5	50	KALIMANTAN TIMUR	SAMARINDA	SUNGAI PINANG	REMAJA	TEMINDUNG PERMAI	PULAU INDAH	34		2023-01-02	10.6	72.5	0	Risiko Lebih	1.71	Normal	0.23	Gizi Lebih	2.14	O	
....
150462	ADNAN IRAGUSTI	L	3	50	KALIMANTAN TIMUR	SAMARINDA	SAMARINDA UTARA	LEMPAKE	LEMPAKE	SRI REJEKI			2023-12-30	3	50		Berat Badan Normal	-0.73	Normal	0.06	Gizi Baik	-1.19	-	
150463	SIENA AL RAISHA RAMADHANI	P	3	52	KALIMANTAN TIMUR	SAMARINDA	SUNGAI PINANG	REMAJA	BANDARA	LESTARI	24		2023-12-13	13	95.5	0	Berat Badan Normal	-1.22	Normal	-1.11	Gizi Baik	-0.85	O	
150464	AFIZAH KHAIRINA	P	2.5	45	KALIMANTAN TIMUR	SAMARINDA	SAMARINDA SEBERANG	MANGKUPALAS	TENUN SAMARINDA	BALO NEGARA	12		2023-12-09	2.5	45		Kurang	-2.03	Pendek	-2.63	Gizi Baik	-0.35	-	
150465	M ARSYA KHOLIF	P	3	49	KALIMANTAN TIMUR	SAMARINDA	SAMARINDA LULU	JUANDA	AIR HITAM	MEKAR SEJAHTERA	32		2023-12-19	3	49		Berat Badan Normal	-1.56	Normal	-1.3	Gizi Baik	-1.04	-	
150466	MUHAMMAD IQBAL	L	2.9	49	KALIMANTAN TIMUR	SAMARINDA	SAMARINDA ILIR	SIDOMULYO	SUNGAI DAMA	RAMANIA	0		2023-12-29	2.9	49		Kurang	-2.97	Pendek	-2.48	Gizi Baik	-1.37	-	

Lampiran 1 Surat Pengantar Pengambilan Data



UMKT
Program Studi
Teknik Informatika
Fakultas Sains dan Teknologi

Telp. 0541-748511 Fax. 0541-766832
Website <http://informatika.umkt.ac.id>
email: informatika@umkt.ac.id

Kalimantan Timur
Berkelanjutan (Environment) Berkemajuan

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Nomor : 003-009/FST.1/D.3/C/2024
Lampiran : -
Perihal : Permohonan Pengambilan Data

Kepada Yth.
Kepala Dinas Kesehatan Kota Samarinda
di -

Tempat

Assalamu'alaikum Warrahmatullahi Wabarrakatuh

Puji Syukur kepada Allah Subhanahu wa ta'ala yang senantiasa melimpahkan Rahmat-Nya kepada kita sekalian. Aamiin.

Schubungan untuk memenuhi Tugas Akhir/Skripsi Tahun Akademik 2023/2024, maka dengan ini kami bermaksud untuk melakukan pengambilan data di Dinas Kesehatan Kota Samarinda. Adapun data yang diminta yaitu data penyakit stunting di Kota Samarinda tahun 2023, dengan nama mahasiswa sebagai berikut:

No	Nama	NIM	Program Studi
1	Ari Ahmad Dhani	2011102441090	Teknik Informatika
2	Bima Satria	2011102441102	Teknik Informatika
3	Lidya Sari	2011102441121	Teknik Informatika
4	Mukminatul Munawaroh	2011102441064	Teknik Informatika
5	Siti Muawwanah	2011102441153	Teknik Informatika

Demikian surat permohonan ini dibuat. Atas perhatiannya dan kerjasamanya kami mengucapkan terima kasih.

Wassalamu'alaikum Warrahmatullahi Wabarrakatuh

Samarinda, 4 Ramadhan 1445 H
15 Maret 2024 M

Program Studi S1 Teknik Informatika










Syaikh, S.Kom., M.TI
NIDN. 1118019203







Lampiran 2 Kartu Bimbingan

KARTU KENDALI BIMBINGAN LAPORAN KARYA ILMIAH

Nama Mahasiswa : Ari Ahmad Dhani
 NIM : 2011102441090
 Nama Dosen Pembimbing : Taghfirul Azhima Yoga Siswa, S.Kom, M.Kom
 Judul Penelitian : PERBAIKAN AKURASI *RANDOM FOREST* DENGAN *ANOVA* DAN *SMOTE* PADA KLASIFIKASI DATA STUNTING

No	Tanggal	Uraian Pembimbingan	Paraf Dosen
1	22 Januari 2024	Membahas tahapan penelitian skripsi	
2	28 Januari 2024	Mencari paper/artikel rujukan penelitian	
3	5 Februari 2024	Review survei paper untuk mencari permasalahan pada klasifikasi data minning dan mecari topik penelitian yg masih relevan untuk diteliti	
4	16 Februari 2024	Review technical paper dan road maps penelitian, untuk mencari algoritma dan metode untuk mengatasi permasalahan klasifikasi	
5	22 Februari 2024	Mencari paper/artikel rujukan yang sesuai dengan topik penelitian yang telah ditentukan	
6	26 Februari 2024	Penentuan judul penelitian	
7	13 Maret 2024	Pembuatan canvas penelitian untuk disubmit ke prodi	
8	15 Maret 2024	Pengajuan surat permohonan data untuk penelitian	
9	18 April 2024	Revisi proposal bab 1, bab 2 dan perbaikan penulisan format skripsi	


10	24 April 2024	Revisi proposal bab 1 dan bab 2, untuk penyelesaian akhir sebelum disubmit	
11	10 Mai 2024	Pembahasan Penulisam bab 3 dan 4 serta naskah publikasi	
12	17 Mai 2024	Revisi naskah skripsi bab 3 dan bab 4 serta naskah publikasi	
13	3 Juni 2024	Revisi naskah Publikasi Sebelum dipublish	


mengetahui

Dosen Pembimbing



Ketua Program Studi


 (Iqbal Fikri Azhima, S.Kom, M.Kom)
 NIDN. 1118038805


 (Arban Syah, S.Kom, M.TI)
 NIDN. 1118019203

RIWAYAT HIDUP



Ari Ahmad Dhani, atau biasa disapa dengan sebutan Ari, lahir di Jawa Tengah tepatnya di daerah Demak pada 07 April 2002, Penulis merupakan anak ke-dua dari Bapak Sukari dan Ibu Rumini. Menempuh pendidikan di SDN 002 Loa Kulu tahun 2008 kemudia 2014, SMPN 1 Loa Kulu tahun 2014 – 2017, SMAN 1 Loa Kulu tahun 2017 – 2020. Kemudian penulis tercatat sebagai mahasiswa pada perguruan tinggi swasta Universitas Muhammadiyah Kalimantan Timur pada Fakultas Sains Dan Teknologi jurusan Teknik informatika pada tahun 2020. Saat menjadi mahasiswa penulis pernah melaksanakan perogram magang di Perusahaan pertambangan batu bara yaitu PT. Bukit Baiduri Energi Kutai Kartainegara selama 3 bulan yang dilaksanakan pada semester 7. Demikian deskripsi riwayat hidup yang penulis sampaikan jika terdapat kesalahan atau kekurangan mohon dimaafkan karena kesempurnaan hanya milik Sang Maha pencipta, maka penulis mengharapkan kritik dan saran mengenai skripsi ini.

JADWAL PENELITIAN

No	Jenis Penelitian	Bulan / 2024						
		Jan	Feb	Mar	Apr	Mei	Jun	Jul
Tahap Pra Penelitian								
1.	Menentukan Judul Penelitian							
2.	Menyusun Rumusan Masalah							
3.	Mencari Data Pendukung							
4.	Menyusun Metode Penelitian							
5.	Menyusun Proposal Penelitian							
6.	Review Desk Simpel							
Tahap Penelitian								
1.	Pengumpulan Data							
2.	Analisis Data							
3.	Pengujian algoritma <i>Random Forest</i>							
Tahap Akhir Penelitian								
1.	Penyusunan Laporan							
2.	Evaluasi							
3.	Penyempurnaan Laporan							
4.	Seminar Hasil							

SELURUH KODE PYTHON

```

import pandas as pd
import numpy as np

from sklearn.model_selection import train_test_split

import seaborn as sns
import matplotlib.pyplot as plt
plt.style.use('seaborn-darkgrid')

```

<ipython-input-1-56fe5db77cb2>:8: MatplotlibDeprecationWarning: The seaborn styles shipped by Matplotlib are deprecated since 3.6, as they no longer correspond to the styles shipped by seaborn. However, plt.style.use('seaborn-darkgrid')

```

[ ] import pandas as pd
data = pd.read_csv('datasetfixstunting.csv')

[ ] data = pd.read_csv('datasetfixstunting.csv')

# Menghitung jumlah data sebelum penghapusan duplikat
total_data_sebelum = len(data)

# Mengurutkan data berdasarkan kolom "Nama" dan "Tanggal Pengukuran" secara menurun
data = data.sort_values(by=['Nama', 'Tanggal Pengukuran'], ascending=[True, False])

# Menghapus duplikat berdasarkan kolom "Nama" dan mempertahankan yang pertama (yang memiliki tanggal pengukuran terbaru)
data = data.drop_duplicates(subset='Nama', keep='first')

# Menghitung jumlah data setelah penghapusan duplikat
total_data_sesudah = len(data)

print(f"Total data sebelum penghapusan duplikat: {total_data_sebelum}")
print(f"Total data setelah penghapusan duplikat: {total_data_sesudah}")

data.head()

```

Total data sebelum penghapusan duplikat: 150465
Total data setelah penghapusan duplikat: 34199

	Nama	JK	Berat	Tinggi	LILA	BB/U	ZS	BB/U	TB/U	ZS	TB/U	BB/TB	ZS	BB/TB	Naik Berat Badan	Jml Vit A	Tanggal Pengukuran
143850	A ALVIN	L	18.06	NaN	18.0	Risiko Lebih	1.19	Normal	0.41	Risiko Gizi Lebih	0.07	N	NaN				2023-12-16
88655	A FADLAN	L	11.06	83.0	0.0	Berat Badan Normal	-0.08	Normal	-0.97	Gizi Baik	0.04	-	NaN				2023-07-10
128459	A FARIS WICAKSONO	L	9.07	78.0	16.0	Berat Badan Normal	-1.75	Pendek	-2.84	Gizi Baik	-0.47	O	1.0				2023-10-06
119828	A FATHAN	L	15.00	107.0	17.0	Berat Badan Normal	-1.08	Normal	0.14	Gizi Baik	-1.83	T	1.0				2023-10-23
20569	A FAUJAN	L	14.00	100.0	0.0	Berat Badan Normal	-0.85	Normal	-0.16	Gizi Baik	-1.14	O	1.0				2023-02-07

```

missing_values = data.isna().sum()

# Menampilkan jumlah missing values untuk setiap kolom
print("Jumlah missing values untuk setiap kolom:")
print(missing_values)

```

Jumlah missing values untuk setiap kolom:

Nama	1
JK	0
Berat	156
Tinggi	11885
LILA	5623
BB/U	0
ZS BB/U	39
TB/U	2640
ZS TB/U	2462
BB/TB	2627
ZS BB/TB	2453
Naik Berat Badan	0
Jml Vit A	21540
Tanggal Pengukuran	0
dtype:	int64

```

data.drop_duplicates(inplace=True)
data.dropna(inplace=True)

pd.set_option('display.max_rows', 10) # Atur jumlah maksimum baris yang ditampilkan
data

```

	Nama	JK	Berat	Tinggi	LILA	BB/U	ZS	BB/U	TB/U	ZS	TB/U	BB/TB	ZS	BB/TB	Naik Berat Badan	Jml Vit A	Tanggal Pengukuran
128459	A FARIS WICAKSONO	L	9.07	78.0	16.0	Berat Badan Normal	-1.75	Pendek	-2.840000	Gizi Baik	-0.47	O	1.0				2023-10-06
119828	A FATHAN	L	15.00	107.0	17.0	Berat Badan Normal	-1.08	Normal	0.140000	Gizi Baik	-1.83	T	1.0				2023-10-23
20569	A FAUJAN	L	14.00	100.0	0.0	Berat Badan Normal	-0.85	Normal	-0.160000	Gizi Baik	-1.14	O	1.0				2023-02-07
31702	A MISHAEL	P	15.00	103.0	0.0	Berat Badan Normal	0.06	Normal	1.050000	Gizi Baik	-0.80	O	1.0				2023-02-09
113633	A ZHAFIR KAMIL	L	14.00	102.0	0.0	Berat Badan Normal	-1.91	Normal	-1.490000	Gizi Baik	-1.59	O	1.0				2023-10-14
...
27525	yelmi	L	17.04	111.0	0.0	Berat Badan Normal	-0.04	Normal	0.060417	Gizi Baik	-0.93	N	1.0				2023-02-27
125185	yumna	P	13.07	100.0	14.0	Berat Badan Normal	-1.07	Normal	-0.450000	Gizi Baik	-1.18	O	1.0				2023-10-14
121521	zidaan aqilla karno	L	14.05	102.0	0.0	Berat Badan Normal	-1.71	Normal	-1.620000	Gizi Baik	-1.15	T	1.0				2023-10-21
22242	zubair	L	17.03	107.0	0.0	Berat Badan Normal	-0.14	Normal	-0.130000	Gizi Baik	-0.13	O	1.0				2023-02-20
112022	zulkifli abdi	L	15.06	102.0	0.0	Berat Badan Normal	-0.59	Normal	-0.690000	Gizi Baik	-0.25	N	1.0				2023-10-03

8059 rows x 14 columns


```
[ ] missing_values2 = data.isnull().sum()

# Menampilkan jumlah missing values untuk setiap kolom
print("Jumlah missing values untuk setiap kolom:")
print(missing_values2)
```

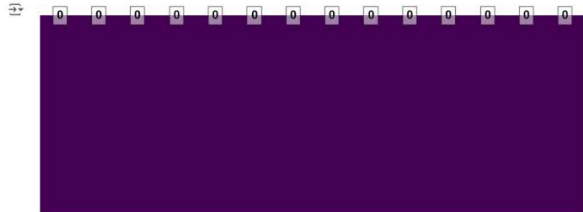
```
Jumlah missing values untuk setiap kolom:
Nama          0
JK            0
Berat         0
Tinggi       0
LiLA         0
BB/TB        0
ZS BB/TB     0
Naik Berat Badan 0
Jml Vit A    0
Tanggal Pengukuran 0
Length: 14, dtype: int64
```

```
[ ] # Buat heatmap untuk menampilkan nilai-nilai yang hilang
plt.figure(figsize=(10, 6)) # Atur ukuran plot
sns.heatmap(data.isna(), yticklabels=False, cbar=False, cmap="viridis")

# Hitung total nilai yang hilang tiap kolom
missing_values_per_column = data.isna().sum()

# Annotate dengan jumlah nilai yang hilang tiap kolom
for i in range(len(missing_values_per_column)):
    plt.text(i + 0.5, -5, f'{missing_values_per_column[i]}', ha='center', va='center',
            color='black', fontsize=12, fontweight='bold', bbox=dict(facecolor='white', alpha=0.5))

plt.show()
```



```
[ ] # Menghapus Kolom Yang Tidak Digunakan
data.drop(columns=["Nama", "Tanggal Pengukuran"], inplace=True)
```

```
[ ] pd.set_option('display_max_rows', 10) # Atur jumlah maksimum baris yang ditampilkan

# Menampilkan DataFrame
data
```

```
JK Berat Tinggi LiLA BB/U ZS BB/U TB/U ZS TB/U BB/TB ZS BB/TB Naik Berat Badan Jml Vit A
```

128459	L	9.07	78.0	16.0	Berat Badan Normal	-1.75	Pendek	-2.840000	Gizi Baik	-0.47	O	1.0
119828	L	15.00	107.0	17.0	Berat Badan Normal	-1.08	Normal	0.140000	Gizi Baik	-1.83	T	1.0
20569	L	14.00	100.0	0.0	Berat Badan Normal	-0.85	Normal	-0.160000	Gizi Baik	-1.14	O	1.0
31702	P	15.00	103.0	0.0	Berat Badan Normal	0.06	Normal	1.050000	Gizi Baik	-0.80	O	1.0
113633	L	14.00	102.0	0.0	Berat Badan Normal	-1.91	Normal	-1.490000	Gizi Baik	-1.59	O	1.0
...
27525	L	17.04	111.0	0.0	Berat Badan Normal	-0.04	Normal	0.060417	Gizi Baik	-0.93	N	1.0
125185	P	13.07	100.0	14.0	Berat Badan Normal	-1.07	Normal	-0.450000	Gizi Baik	-1.18	O	1.0
121521	L	14.05	102.0	0.0	Berat Badan Normal	-1.71	Normal	-1.620000	Gizi Baik	-1.15	T	1.0
22242	L	17.03	107.0	0.0	Berat Badan Normal	-0.14	Normal	-0.130000	Gizi Baik	-0.13	O	1.0
112022	L	15.06	102.0	0.0	Berat Badan Normal	-0.59	Normal	-0.690000	Gizi Baik	-0.25	N	1.0

8059 rows x 12 columns

```
[ ] from sklearn.preprocessing import LabelEncoder

# Buat instance LabelEncoder
encoder = LabelEncoder()

# Kolom yang perlu di-encode
columns_to_encode = ['JK', 'BB/U', 'BB/TB', 'Naik Berat Badan']

# Loop melalui setiap kolom yang perlu di-encode dan terapkan LabelEncoder
for column in columns_to_encode:
    data[column] = encoder.fit_transform(data[column])

# Menampilkan hasilnya untuk memastikan encoding telah berhasil
pd.set_option('display_max_rows', 10) # Atur jumlah maksimum baris yang ditampilkan

# Menampilkan DataFrame
data
```

```
JK Berat Tinggi LiLA BB/U ZS BB/U TB/U ZS TB/U BB/TB ZS BB/TB Naik Berat Badan Jml Vit A
```

128459	0	9.07	78.0	16.0	0	-1.75	Pendek	-2.840000	0	-0.47	2	1.0
119828	0	15.00	107.0	17.0	0	-1.08	Normal	0.140000	0	-1.83	3	1.0
20569	0	14.00	100.0	0.0	0	-0.85	Normal	-0.160000	0	-1.14	2	1.0
31702	1	15.00	103.0	0.0	0	0.06	Normal	1.050000	0	-0.80	2	1.0
113633	0	14.00	102.0	0.0	0	-1.91	Normal	-1.490000	0	-1.59	2	1.0
...
27525	0	17.04	111.0	0.0	0	-0.04	Normal	0.060417	0	-0.93	1	1.0
125185	1	13.07	100.0	14.0	0	-1.07	Normal	-0.450000	0	-1.18	2	1.0
121521	0	14.05	102.0	0.0	0	-1.71	Normal	-1.620000	0	-1.15	3	1.0
22242	0	17.03	107.0	0.0	0	-0.14	Normal	-0.130000	0	-0.13	2	1.0
112022	0	15.06	102.0	0.0	0	-0.59	Normal	-0.690000	0	-0.25	1	1.0

8059 rows x 12 columns


```

# Ubah TB/U
data['TB/U'] = data['TB/U'].replace({'Normal': 0, 'Tinggi': 0, 'Pendek': 1, 'Sangat Pendek': 1})

pd.set_option('display.max_rows', 10) # Atur jumlah maksimum baris yang ditampilkan

data

JK Berat Tinggi LiLA BB/U ZS BB/U TB/U ZS TB/U BB/TB ZS BB/TB Naik Berat Badan Jml Vit A
128459 0 9.07 78.0 16.0 0 -1.75 1 -2.840000 0 -0.47 2 1.0
119828 0 15.00 107.0 17.0 0 -1.08 0 0.140000 0 -1.83 3 1.0
20569 0 14.00 100.0 0.0 0 -0.85 0 -0.160000 0 -1.14 2 1.0
31702 1 15.00 103.0 0.0 0 0.06 0 1.050000 0 -0.80 2 1.0
113633 0 14.00 102.0 0.0 0 -1.91 0 -1.490000 0 -1.59 2 1.0
...
27525 0 17.04 111.0 0.0 0 -0.04 0 0.060417 0 -0.93 1 1.0
125185 1 13.07 100.0 14.0 0 -1.07 0 -0.450000 0 -1.18 2 1.0
121521 0 14.05 102.0 0.0 0 -1.71 0 -1.620000 0 -1.15 3 1.0
22242 0 17.03 107.0 0.0 0 -0.14 0 -0.130000 0 -0.13 2 1.0
112022 0 15.06 102.0 0.0 0 -0.59 0 -0.690000 0 -0.25 1 1.0
8059 rows x 12 columns

data.info()
<class 'pandas.core.frame.DataFrame'>
Index: 8059 entries, 128459 to 112022
Data columns (total 12 columns):
# Column Non-Null Count Dtype
---
0 JK 8059 non-null int64
1 Berat 8059 non-null float64
2 Tinggi 8059 non-null float64
3 LiLA 8059 non-null float64
4 BB/U 8059 non-null int64
5 ZS BB/U 8059 non-null float64
6 TB/U 8059 non-null int64
7 ZS TB/U 8059 non-null float64
8 BB/TB 8059 non-null int64
9 ZS BB/TB 8059 non-null float64
10 Naik Berat Badan 8059 non-null int64
11 Jml Vit A 8059 non-null float64
dtypes: float64(7), int64(5)
memory usage: 1.1 MB

x = data.drop(['TB/U'], axis=1)
y = data['TB/U']

pd.set_option('display.max_rows', 10)

data

JK Berat Tinggi LiLA BB/U ZS BB/U TB/U ZS TB/U BB/TB ZS BB/TB Naik Berat Badan Jml Vit A
128459 0 9.07 78.0 16.0 0 -1.75 1 -2.840000 0 -0.47 2 1.0
119828 0 15.00 107.0 17.0 0 -1.08 0 0.140000 0 -1.83 3 1.0
20569 0 14.00 100.0 0.0 0 -0.85 0 -0.160000 0 -1.14 2 1.0
31702 1 15.00 103.0 0.0 0 0.06 0 1.050000 0 -0.80 2 1.0
113633 0 14.00 102.0 0.0 0 -1.91 0 -1.490000 0 -1.59 2 1.0
...
27525 0 17.04 111.0 0.0 0 -0.04 0 0.060417 0 -0.93 1 1.0
125185 1 13.07 100.0 14.0 0 -1.07 0 -0.450000 0 -1.18 2 1.0
121521 0 14.05 102.0 0.0 0 1.71 0 1.620000 0 1.15 3 1.0
22242 0 17.03 107.0 0.0 0 -0.14 0 -0.130000 0 -0.13 2 1.0
112022 0 15.06 102.0 0.0 0 -0.59 0 -0.690000 0 -0.25 1 1.0
8059 rows x 12 columns

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
x = scaler.fit_transform(x)

x
array([[0. , 0.29523676, 0.4 , ..., 0.46683333, 0.66666667,
0. ],
[0. , 0.49491487, 0.81428571, ..., 0.3475 , 1. ,
0. ],
[0. , 0.46124167, 0.71428571, ..., 0.485 , 0.66666667,
0. ],
...,
[0. , 0.46292529, 0.74285714, ..., 0.40416667, 1. ,
0. ],
[0. , 0.56326903, 0.81428571, ..., 0.48916667, 0.66666667,
0. ],
...])

```

```
[ ] data['TB/U'].value_counts()
```

```
TB/U
0    7317
1     742
Name: count, dtype: int64
```

```
[ ] class_data = data['TB/U'].value_counts()

print('Jumlah data kelas Stunting:', class_data[1])
print('Jumlah data kelas Tidak Stunting:', class_data[0])

# Create a bar plot
sns.barplot(x=class_data.index, y=class_data.values, palette=['orange', 'steelblue'])
plt.title("Jumlah Data Tidak Stunting dan Stunting")
plt.xlabel("Kelas")
plt.ylabel("Jumlah Data")
plt.show()
```

```
Jumlah data kelas Stunting: 742
```

```
Jumlah data kelas Tidak Stunting: 7317
```

```
<ipython-input-19-1da1770b606e>:7: FutureWarning:
```

Passing 'palette' without assigning 'hue' is deprecated and will be removed in v0.14.0. Assign the 'x' variable to 'hue' and set 'legend=False' for the same effect.

```
sns.barplot(x=class_data.index, y=class_data.values, palette=['orange', 'steelblue'])
```



```
[ ] x = data.drop(['TB/U'], axis=1)
y = data['TB/U']
from imblearn.over_sampling import SMOTE
sm = SMOTE(random_state=42, sampling_strategy=1)
X_inb, y_inb = sm.fit_resample(x, y)
```

```
[ ] class_data = y_inb.value_counts()

print('Jumlah data kelas stunting:', class_data[1])
print('Jumlah data kelas tidak stunting:', class_data[0])

# Create a bar plot
sns.barplot(x=class_data.index, y=class_data.values, palette=['orange', 'steelblue'])
plt.title("Jumlah Data Tidak Stunting dan Stunting")
plt.xlabel("Kelas")
plt.ylabel("Jumlah Data")
plt.show()
```

```
Jumlah data kelas stunting: 7317
```

```
Jumlah data kelas tidak stunting: 7317
```

```
<ipython-input-21-3c98e40829bf>:7: FutureWarning:
```

Passing 'palette' without assigning 'hue' is deprecated and will be removed in v0.14.0. Assign the 'x' variable to 'hue' and set 'legend=False' for the same effect.

```
sns.barplot(x=class_data.index, y=class_data.values, palette=['orange', 'steelblue'])
```



```

from sklearn.model_selection import cross_val_score, StratifiedKFold
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score, f1_score
import numpy as np

# Perform cross-validation on the Random Forest Classifier with K=10
clf = RandomForestClassifier(max_depth=1, random_state=42)

# Kode kfold Cross Validation
cv = StratifiedKFold(n_splits=10, shuffle=True, random_state=42)

# Lists to store evaluation metrics for each fold
conf_matrices = []
accuracies = []
precisions = []
recalls = []
f1_scores = []

for train_index, test_index in cv.split(X_1mb, y_1mb):
    X_train, X_test = X_1mb.iloc[train_index], X_1mb.iloc[test_index]
    y_train, y_test = y_1mb.iloc[train_index], y_1mb.iloc[test_index]

    # Train the model
    clf.fit(X_train, y_train)

    # Predict on the test set
    y_pred = clf.predict(X_test)

    # Calculate evaluation metrics
    conf_matrices.append(confusion_matrix(y_test, y_pred))
    accuracies.append(accuracy_score(y_test, y_pred))

# Print the evaluation metrics for each fold
for i, (conf_matrix, accuracy) in enumerate(zip(conf_matrices, accuracies), 1):
    print(f"Fold-{i}: Accuracy={accuracy}")

# Print the average evaluation metrics
print("")
print('Hasil Rata Rata')
print(f"Average Accuracy: {np.mean(accuracies)}")

# Calculate and print the average confusion matrix
avg_conf_matrix = sum(conf_matrices)
print("Average Confusion Matrix:")
print(avg_conf_matrix)

```

```

Fold-1: Accuracy=0.9795981967213115
Fold-2: Accuracy=0.9883879781420765
Fold-3: Accuracy=0.9898718382513661
Fold-4: Accuracy=0.9898718382513661
Fold-5: Accuracy=0.9856459330143541
Fold-6: Accuracy=0.9917976760082023
Fold-7: Accuracy=0.9938482570061518
Fold-8: Accuracy=0.9890635688109364
Fold-9: Accuracy=0.9917976760082023
Fold-10: Accuracy=0.9849624060150376

Hasil Rata Rata
Average Accuracy: 0.9883153767429006
Average Confusion Matrix:
[[ 7178  139]
 [   32  7285]]

```

```

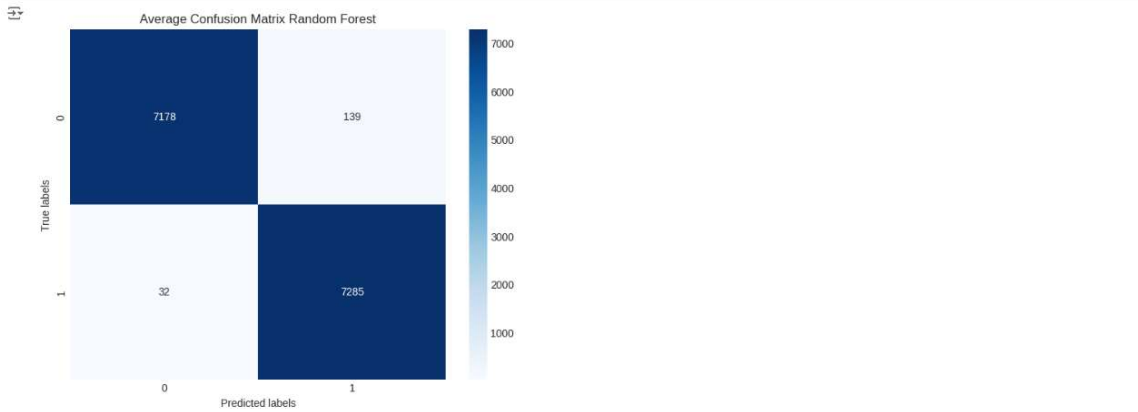
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# Calculate the average confusion matrix
avg_conf_matrix = sum(conf_matrices)

# Create a heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(avg_conf_matrix, annot=True, fmt=".0f", cmap="Blues")

plt.xlabel("Predicted labels")
plt.ylabel("True labels")
plt.title("Average Confusion Matrix Random Forest")
plt.show()

```



```
data
```

	JK	Berat	Tinggi	LiLA	BB/U	ZS	BB/U	TB/U	ZS	TB/U	BB/TB	ZS	BB/TB	Naik Berat Badan	Jml Vit A
128459	0	9.07	78.0	16.0	0	-1.75	1	-2.840000	0	-0.47				2	1.0
119828	0	15.00	107.0	17.0	0	-1.08	0	0.140000	0	-1.83				3	1.0
20569	0	14.00	100.0	0.0	0	-0.85	0	-0.160000	0	-1.14				2	1.0
31702	1	15.00	103.0	0.0	0	0.06	0	1.050000	0	-0.80				2	1.0
113633	0	14.00	102.0	0.0	0	-1.91	0	-1.490000	0	-1.59				2	1.0
...
27525	0	17.04	111.0	0.0	0	-0.04	0	0.060417	0	-0.93				1	1.0
125185	1	13.07	100.0	14.0	0	-1.07	0	-0.450000	0	-1.18				2	1.0
121521	0	14.05	102.0	0.0	0	-1.71	0	-1.620000	0	-1.15				3	1.0
22242	0	17.03	107.0	0.0	0	-0.14	0	-0.130000	0	-0.13				2	1.0
112022	0	15.06	102.0	0.0	0	-0.59	0	-0.690000	0	-0.25				1	1.0

8059 rows x 12 columns

```
[ ] from sklearn.feature_selection import SelectKBest, f_classif

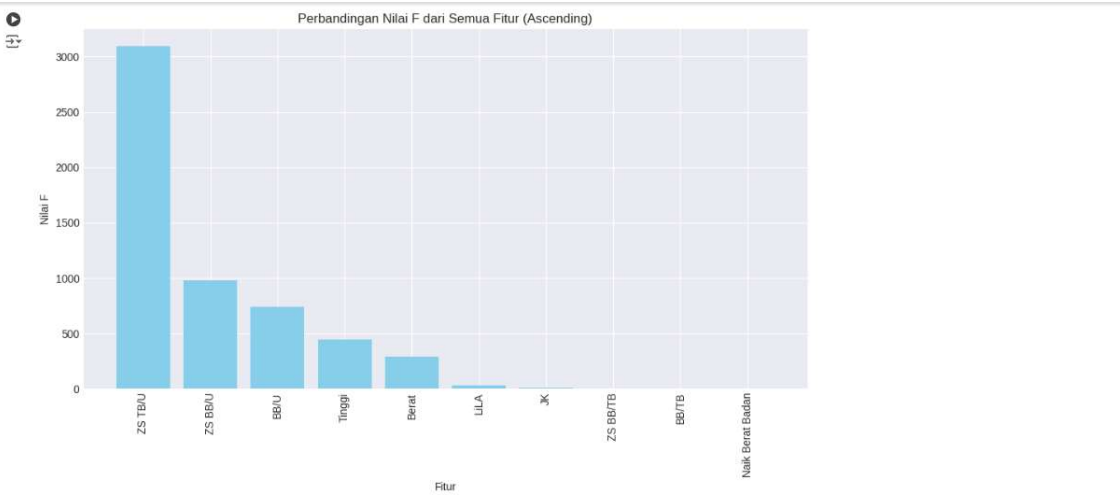
# Menggunakan SelectKBest dengan ANOVA untuk memilih fitur terbaik
selector = SelectKBest(score_func=f_classif)
X_selected = selector.fit_transform(x, y)

# Mengambil nilai F dan nama fitur dari SelectKBest
f_values = selector.scores_
features = x.columns

# Membuat DataFrame dari fitur dan nilai F
import pandas as pd
df_f = pd.DataFrame({'Feature': features, 'F_Value': f_values})

# Sorting DataFrame berdasarkan nilai F secara ascending
df_sorted = df_f.sort_values('F_Value', ascending=False)

# Membuat diagram bar untuk menampilkan nilai F dari semua fitur
plt.figure(figsize=(12, 6))
plt.bar(df_sorted['Feature'], df_sorted['F_Value'], color='skyblue')
plt.xlabel('Fitur')
plt.ylabel('Nilai F')
plt.title('Perbandingan Nilai F dari Semua Fitur (Ascending)')
plt.xticks(rotation=90)
plt.show()
```



```
[ ] # Mengatur lebar output konsol
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('display.width', None)

# Print nilai F dari semua fitur
print(df_sorted[['Feature', 'F_Value']])
```

```
[ ]
```

	Feature	F_Value
6	ZS TB/U	3091.918974
5	ZS BB/U	975.781238
4	BB/U	737.297857
2	Tinggi	441.561417
1	Berat	290.194343
3	LiLA	32.357488
0	JK	8.574745
8	ZS BB/TB	4.964478
7	BB/TB	4.669220
9	Naik Berat Badan	3.936491
10	Jml Vit A	NaN

```
[ ] # Mengembalikan pengaturan lebar output konsol seperti semula
pd.reset_option('display.max_rows')
pd.reset_option('display.max_columns')
pd.reset_option('display.width')
```

```

# Seleksi Fitur P-Value < 0,1
x = data.drop(['Haik Berat Badan', 'BB/TB', 'TB/U', '3ml Vit A', 'ZS BB/TB', 'JK'], axis=1)
y = data['TB/U']
from imblearn.over_sampling import SMOTE
sm = SMOTE(random_state=42, sampling_strategy=1)
X_Anv, y_Anv = sm.fit_resample(x, y)

```

[] x

	Berat	Tinggi	LiLA	BB/U	ZS	BB/U	ZS	TB/U
128459	9.07	78.0	16.0	0	-1.75	-2.840000		
119828	15.00	107.0	17.0	0	-1.08	0.140000		
20569	14.00	100.0	0.0	0	-0.85	-0.160000		
31702	15.00	103.0	0.0	0	0.06	1.050000		
113633	14.00	102.0	0.0	0	-1.91	-1.490000		
...
27525	17.04	111.0	0.0	0	-0.04	0.060417		
125185	13.07	100.0	14.0	0	-1.07	-0.450000		
121521	14.05	102.0	0.0	0	-1.71	-1.620000		
22242	17.03	107.0	0.0	0	-0.14	-0.130000		
112022	15.06	102.0	0.0	0	-0.59	-0.690000		

8059 rows x 6 columns

```

[ ] from sklearn.model_selection import cross_val_score, StratifiedKFold
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score, f1_score
import numpy as np

# Perform cross-validation on the Random Forest classifier with K=10
clf = RandomForestClassifier(max_depth=1, random_state=42)

# Kode kfold Cross Validation
cv = StratifiedKFold(n_splits=10, shuffle=True, random_state=42)

# Lists to store evaluation metrics for each fold
conf_matrices = []
accuracies = []

for train_index, test_index in cv.split(X_Anv, y_Anv):
    X_train, X_test = X_Anv.iloc[train_index], X_Anv.iloc[test_index]
    y_train, y_test = y_Anv.iloc[train_index], y_Anv.iloc[test_index]

    # Train the model
    clf.fit(X_train, y_train)

    # Predict on the test set
    y_pred = clf.predict(X_test)

    # Calculate evaluation metrics
    # Calculate evaluation metrics
    conf_matrices.append(confusion_matrix(y_test, y_pred))
    accuracies.append(accuracy_score(y_test, y_pred))

# Print the evaluation metrics for each fold
for i, (conf_matrix, accuracy) in enumerate(zip(conf_matrices, accuracies), 1):
    print(f"Fold-{i}: Accuracy={accuracy}")

# Print the average evaluation metrics
print("")
print('Hasil Rate Rate')
print(f"Average Accuracy: {np.mean(accuracies)}")

# Calculate and print the average confusion matrix
avg_conf_matrix = sum(conf_matrices)
print("Average Confusion Matrix:")
print(avg_conf_matrix)

```

```

Fold-1: Accuracy=0.9965846994535519
Fold-2: Accuracy=0.9979508196721312
Fold-3: Accuracy=0.9986338797814208
Fold-4: Accuracy=0.9993169398907104
Fold-5: Accuracy=0.9965823650034177
Fold-6: Accuracy=0.9979494190020506
Fold-7: Accuracy=0.9979494190020506
Fold-8: Accuracy=0.9952153110047847
Fold-9: Accuracy=0.9986329460013671
Fold-10: Accuracy=0.9986329460013671

Hasil Rate Rate
Average Accuracy: 0.9977448744812852
Average Confusion Matrix:
[[7317  0]
 [ 33 7284]]

```

```
[ ] import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# Calculate the average confusion matrix
avg_conf_matrix = sum(conf_matrices)

# Create a heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(avg_conf_matrix, annot=True, fmt=".0f", cmap="Blues")

plt.xlabel("Predicted labels")
plt.ylabel("True labels")
plt.title("Average Confusion Matrix Random Forest")
plt.show()
```

