

**LAPORAN**

**RENCANA TUGAS MAHASISWA (RTM) Ke-1**

**MATA KULIAH ANALISIS DATA EKSPLORATIF**

**“PENYAJIAN DATA MENGGUNAKAN BERBAGAI**

**BENTUK DIAGRAM”**



**DISUSUN OLEH:**

Muhammad Aryasatya Nugroho ( 22083010085 )

**DOSEN PENGAMPU:**

Tresna Maulana Fahrudin S.ST., M.T. (NIP. 199305012022031007)

**PROGRAM STUDI SAINS DATA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN” JAWA TIMUR**

**2023**

Sajikanlah dataset berikut ini ke dalam bentuk diagram batang (bar chart), diagram garis (line chart), diagram pencar (scatter chart), diagram lingkaran (pie chart), diagram area, atau diagram lainnya yang cocok untuk merepresentasikan dataset. Lakukan praproses atau penyesuaian atribut/kolom untuk memudahkan dalam penyajian dan mendapatkan insight yang bermakna.

### 1. Daftar Dosen UPN Veteran X Angkatan 2022

No.	Tanggal Lahir	Pendidikan Terakhir
1.	1 Mei 1993	S2 Terapan Teknik Informatika, Politeknik Elektronika Negeri Surabaya
2.	8 September 1992	S2 Teknik Informatika, Institut Teknologi Sepuluh Nopember
3.	4 Agustus 1978	S3 Biomedical Engineering, University of Rome
4.	1 Januari 1980	S3 Informatics, University of WDaftashington
5.	16 April 1995	S3 Biomedik, Institut Teknologi Sepuluh Nopember
6.	19 Maret 1991	S3 Informatika, Universitas Telkom
7.	23 November 1994	S2 Terapan Teknik Elektronika, Politeknik Elektronika Negeri Surabaya
8.	25 Februari 1997	S1 Electronical Engineering, University of Manchester
9.	20 Oktober 1991	S3 Informatics, University of Nigeria
10.	18 November 1997	S1 Biomedik, Universitas Indonesia
11.	11 Mei 1992	S3 Teknik Elektro, Institut Teknologi Sepuluh Nopember
12.	12 Juni 1982	Post-doctoral Biomedical Engineering, Pusan National University
13.	11 Juli 1979	Post-doctoral Electronical Engineering, Hongkong University
14.	16 Agustus 1977	Post-doctoral Biomedical Engineering, Sydney University
15.	2 November 1970	Post-doctoral Informatics, Institut Teknologi Bandung
16.	8 Desember 1983	S3 Sistem Informasi, Institut Teknologi Sepuluh Nopember
17.	22 Maret 1992	S2 Teknik Informatika, Universitas Padjadjaran
18.	10 April 1994	S2 Teknik Elektro, Universitas Gadjah Mada
19.	12 Mei 1996	S2 Teknik Biomedik, Universitas Airlangga
20.	11 April 1993	S2 Teknik Informatika, Universitas Bina Darma

#### a) Import Library yang dibutuhkan

```

1 import pandas as pd
2 from datetime import datetime
3 import locale
4 import seaborn as sns

```

Kode script diatas merupakan contoh penggunaan Python dengan library seperti Pandas, Seaborn, DateTime untuk analisis data.

```

4 data = {
5     'Nomor': list(range(1, 21)), # Nomor dosen
6     'Tanggal Lahir': [
7         '1 Mei 1993', '8 September 1992', '4 Agustus 1978', '1 Januari 1980', '16 April 1995',
8         '19 Maret 1991', '23 November 1994', '25 Februari 1997', '20 Oktober 1991', '18 November 1997',
9         '11 Mei 1992', '12 Juni 1982', '11 Juli 1979', '16 Agustus 1977', '2 November 1970',
10        '8 Desember 1983', '22 Maret 1992', '10 April 1994', '12 Mei 1996', '11 April 1993'
11    ],
12     'Pendidikan Terakhir': [
13         'S2 Terapan Teknik Informatika, Politeknik Elektronika Negeri Surabaya',
14         'S2 Teknik Informatika, Institut Teknologi Sepuluh Nopember',
15         'S3 Biomedical Engineering, University of Rome',
16         'S3 Informatics, University of Wdashington',
17         'S3 Biomedik, Institut Teknologi Sepuluh Nopember',
18         'S3 Informatika, Universitas Telkom',
19         'S2 Terapan Teknik Elektronika, Politeknik Elektronika Negeri Surabaya',
20         'S1 Electronical Engineering, University of Manchester',
21         'S3 Informatics, University of Nigeria',
22         'S1 Biomedik, Universitas Indonesia',
23         'S3 Teknik Elektro, Institut Teknologi Sepuluh Nopember',
24         'Post-doctoral Biomedical Engineering, Pusan National University',
25         'Post-doctoral Electronical Engineering, Hongkong University',
26         'Post-doctoral Biomedical Engineering, Sydney University',
27         'Post-doctoral Informatics, Institut Teknologi Bandung',
28         'S3 Sistem Informasi, Institut Teknologi Sepuluh Nopember',
29         'S2 Teknik Informatika, Universitas Padjadjaran',
30         'S2 Teknik Elektro, Universitas Gadjah Mada',
31         'S2 Teknik Biomedik, Universitas Airlangga',
32         'S2 Teknik Informatika, Universitas Bina Darma'
33     ]
34 }
35
36 df = pd.DataFrame(data)
37 df

```

Data diatas berisi informasi tentang dosen-dosen dengan berbagai atribut seperti nomor identifikasi, tanggal lahir, dan Pendidikan terakhir. Dataset ini bertujuan untuk melakukan analisis tentang profil dosen, seperti distribusi usia atau jenjang Pendidikan terakhir. DataFrame dibuat menggunakan library Pandas untuk mengorganisir data secara efisien, dan memungkinkan untuk melakukan berbagai operasi analisis dan visualisasi data yang diperlukan. Output DataFrame:

	Nomor	Tanggal Lahir	Pendidikan Terakhir
0	1	1 Mei 1993	S2 Terapan Teknik Informatika, Politeknik Elek...
1	2	8 September 1992	S2 Teknik Informatika, Institut Teknologi Sepu...
2	3	4 Agustus 1978	S3 Biomedical Engineering, University of Rome
3	4	1 Januari 1980	S3 Informatics, University of Wdashington
4	5	16 April 1995	S3 Biomedik, Institut Teknologi Sepuluh Nopember
5	6	19 Maret 1991	S3 Informatika, Universitas Telkom
6	7	23 November 1994	S2 Terapan Teknik Elektronika, Politeknik Elek...
7	8	25 Februari 1997	S1 Electronical Engineering, University of Man...
8	9	20 Oktober 1991	S3 Informatics, University of Nigeria
9	10	18 November 1997	S1 Biomedik, Universitas Indonesia
10	11	11 Mei 1992	S3 Teknik Elektro, Institut Teknologi Sepuluh ...
11	12	12 Juni 1982	Post-doctoral Biomedical Engineering, Pusan Na...
12	13	11 Juli 1979	Post-doctoral Electronical Engineering, Hongko...
13	14	16 Agustus 1977	Post-doctoral Biomedical Engineering, Sydney U...
14	15	2 November 1970	Post-doctoral Informatics, Institut Teknologi ...
15	16	8 Desember 1983	S3 Sistem Informasi, Institut Teknologi Sepulu...
16	17	22 Maret 1992	S2 Teknik Informatika, Universitas Padjadjaran
17	18	10 April 1994	S2 Teknik Elektro, Universitas Gadjah Mada
18	19	12 Mei 1996	S2 Teknik Biomedik, Universitas Airlangga
19	20	11 April 1993	S2 Teknik Informatika, Universitas Bina Darma

Output DataFrame diatas mencakup informasi tentang 20 dosen, dengan atribut yang mencakup nomor identifikasi, tanggal lahir, dan pendidikan terakhir. Selanjutnya akan dilakukan penyesuaian kolom agar dapat memahami profil dosen ini lebih baik, seperti

menentukan distribusi usia dosen, mencari tahu pendidikan terakhir yang paling umum, serta insight lainnya.

```

1 df = pd.DataFrame(data)
2
3 # formatting tanggal
4 locale.setlocale(locale.LC_TIME, 'id_ID.UTF-8')
5 df['Tanggal Lahir'] = pd.to_datetime(df['Tanggal Lahir'], format='%d %B %Y', errors='coerce')
6
7 # jenjang pendidikan
8 df['Jenjang Pendidikan'] = df['Pendidikan Terakhir'].str.extract(r'(S\d|Post-doctoral)')[0]
9
10 # jumlah dosen per jenjang pendidikan
11 jumlah_dosen_per_jenjang = df['Jenjang Pendidikan'].value_counts().reset_index()
12 jumlah_dosen_per_jenjang.columns = ['Jenjang Pendidikan', 'Jumlah Dosen']
13
14 # bidang studi
15 df['Bidang Studi'] = df['Pendidikan Terakhir'].str.replace(r'(S\d|Post-doctoral)', '').str.split(',').str[0].str.strip()
16 df['Kampus'] = df['Pendidikan Terakhir'].str.replace(r'(S\d|Post-doctoral)', '').str.split(',').str[1].str.strip()
17
18 df['Usia Sekarang'] = (datetime.now() - df['Tanggal Lahir']).astype('<m8[Y]')
19
20 # range umur
21 bins = [0, 30, 40, 50, 60, 70, float('inf')]
22 labels = ['<30', '30-40', '41-50', '51-60', '61-70', '70+']
23 df['Rentang Umur'] = pd.cut(df['Usia Sekarang'], bins=bins, labels=labels)
24
25 # Menghapus kolom-kolom yang tidak diperlukan
26 df = df.drop(columns=['Nomor', 'Pendidikan Terakhir'])
27 df

```

Kode diatas digunakan untuk melakukan transformasi data yang berisi informasi tentang dosen. Sepeerti mengubah format tanggal lahir menjadi sesuai, kemudian mengekstrak jenjang pendidikan dari kolom pendidikan terakhir dan memisahkan nama kampus. Selain itu, dilakukan perhitungan usia dosen berdasarkan tanggal lahir mereka dan mengelompokkan mereka ke dalam rentang usia. Transformasi ini akan membantu dan memudahkan analisis tentang profil dosen. Output DataFrame setelah penyesuaian:

	Tanggal Lahir	Jenjang Pendidikan	Bidang Studi		Kampus	Usia Sekarang	Rentang Umur
0	1993-05-01	S2	Terapan Teknik Informatika	Politeknik Elektronika Negeri Surabaya		30.0	<30
1	1992-09-08	S2	Teknik Informatika	Institut Teknologi Sepuluh Nopember		31.0	30-40
2	1978-08-04	S3	Biomedical Engineering	University of Rome		45.0	41-50
3	1980-01-01	S3	Informatics	University of WDaashington		43.0	41-50
4	1995-04-16	S3	Biomedik	Institut Teknologi Sepuluh Nopember		28.0	<30
5	1991-03-19	S3	Informatika	Universitas Telkom		32.0	30-40
6	1994-11-23	S2	Terapan Teknik Elektronika	Politeknik Elektronika Negeri Surabaya		28.0	<30
7	1997-02-25	S1	Electronical Engineering	University of Manchester		26.0	<30
8	1991-10-20	S3	Informatics	University of Nigeria		31.0	30-40
9	1997-11-18	S1	Biomedik	Universitas Indonesia		25.0	<30
10	1992-05-11	S3	Teknik Elektro	Institut Teknologi Sepuluh Nopember		31.0	30-40
11	1982-06-12	Post-doctoral	Biomedical Engineering	Pusan National University		41.0	41-50
12	1979-07-11	Post-doctoral	Electronical Engineering	Hongkong University		44.0	41-50
13	1977-08-16	Post-doctoral	Biomedical Engineering	Sydney University		46.0	41-50
14	1970-11-02	Post-doctoral	Informatics	Institut Teknologi Bandung		52.0	51-60
15	1983-12-08	S3	Sistem Informasi	Institut Teknologi Sepuluh Nopember		39.0	30-40
16	1992-03-22	S2	Teknik Informatika	Universitas Padjadjaran		31.0	30-40
17	1994-04-10	S2	Teknik Elektro	Universitas Gajah Mada		29.0	<30
18	1996-05-12	S2	Teknik Biomedik	Universitas Airlangga		27.0	<30
19	1993-04-11	S2	Teknik Informatika	Universitas Bina Darma		30.0	<30

Setelah dilakukan penyesuaian kolom data, maka kita dapat lebih mudah melakukan beberapa visualisasi data dan menemukan insight data.

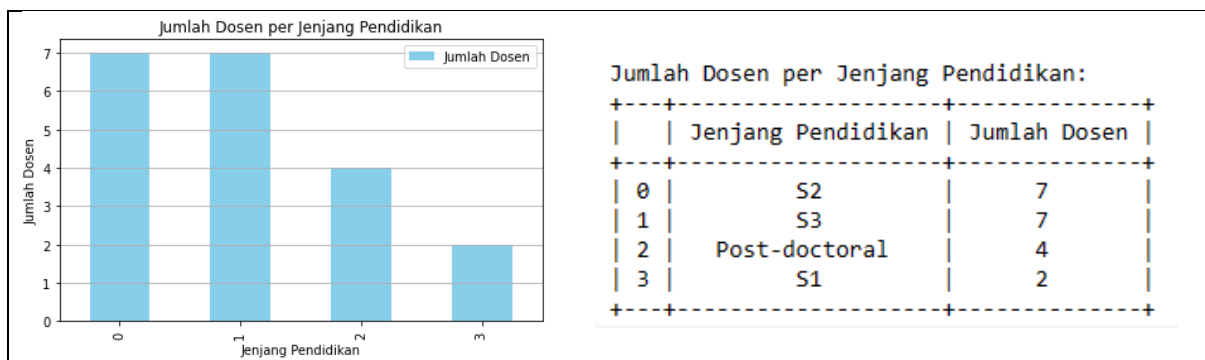
#### a) Visualisasi Bar Chart

```

1 # Membuat bar chart
2 plt.figure(figsize=(10, 6))
3 jumlah_dosen_per_jenjang.plot(kind='bar', color='skyblue')
4 plt.title('Jumlah Dosen per Jenjang Pendidikan')
5 plt.xlabel('Jenjang Pendidikan')
6 plt.ylabel('Jumlah Dosen')
7 plt.grid(axis='y')
8
9 # visualisasi
10 plt.tight_layout()
11 plt.show()
12
13 # format tabel
14 print("Jumlah Dosen per Jenjang Pendidikan:")
15 print(tabulate(jumlah_dosen_per_jenjang.reset_index(), headers=['Jenjang Pendidikan', 'Jumlah Dosen'], tablefmt='pretty')

```

Output bar chart yang dihasilkan menampilkan informasi tentang jumlah dosen per jenjang pendidikan dalam visual yang jelas. Output:



Dari grafik diatas, dapat dengan mudah dilihat bahwa jenjang pendidikan S2 memiliki jumlah dosen terbanyak, diikuti oleh S3.. Grafik ini memberikan gambar visual mengenai distribusi jenjang pendidikan dosen dalam suatu institusi pendidikan, memudahkan pemahaman struktur pendidikan dosen.

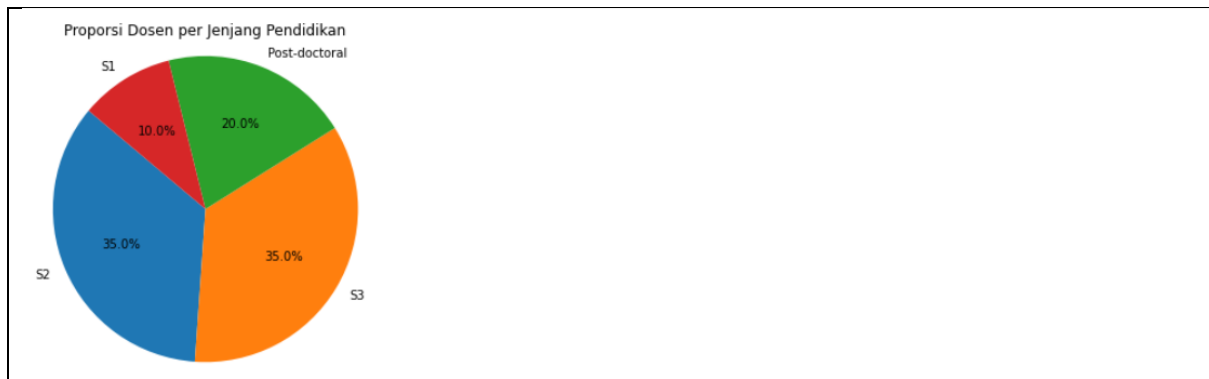
## b) Pie Chart

```

1 # Membuat pie chart
2 plt.figure(figsize=(8, 8))
3 plt.pie(jumlah_dosen_per_jenjang, labels=jumlah_dosen_per_jenjang.index, autopct='%1.1f%%', startangle=140)
4 plt.title('Proporsi Dosen per Jenjang Pendidikan')
5 plt.axis('equal')
6
7 # Tampilkan visualisasi
8 plt.show()

```

Visualisasi pie chart diatas memberikan gambaran tentang proporsi jenjang pendidikan dalam bentuk lingkaran. Setiap bagian dari lingkaran mewakili presentasi dosen yang memiliki jenjang pendidikan tertentu dalam dataset.



Label-label dalam lingkaran menunjukkan jenjang pendidikan, dan angka-angka yang ditampilkan di sekitar lingkaran adalah presentasi total dari jumlah dosen yang dimiliki oleh setiap jenjang pendidikan.

Dalam Perbandingan dengan diagram batang, pie chart lebih tepat untuk menggambarkan proporsi atau presentase dalam dataset, sedangkan bar chart lebih condong untuk membandingkan jumlah absolut antara kategori.

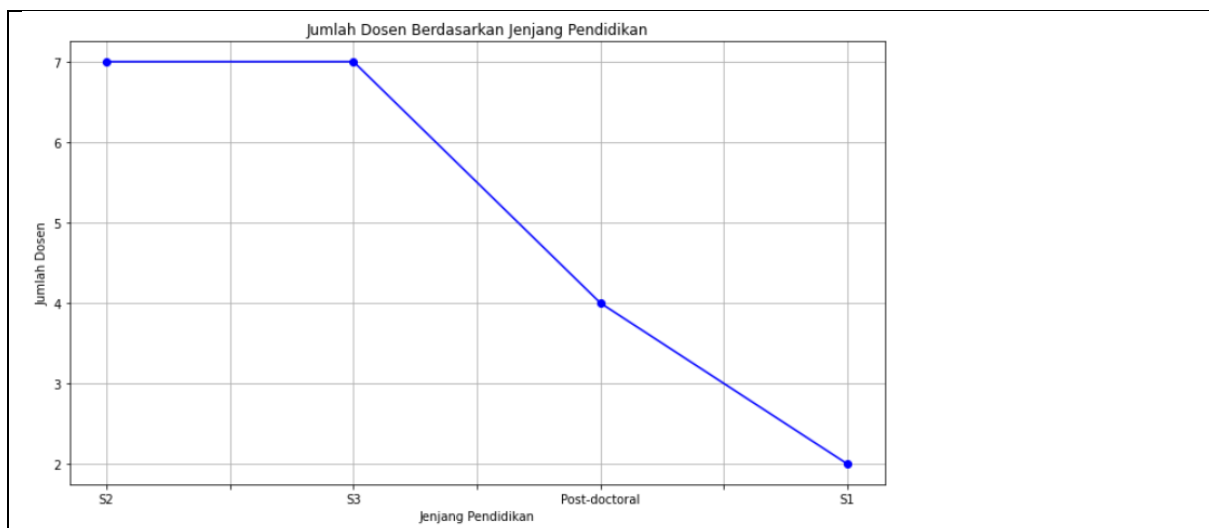
### c) Line Chart

```

1 # Membuat line chart
2 plt.figure(figsize=(10, 6))
3 jumlah_dosen_per_jenjang.plot(kind='line', marker='o', color='blue')
4 plt.title('Jumlah Dosen Berdasarkan Jenjang Pendidikan')
5 plt.xlabel('Jenjang Pendidikan')
6 plt.ylabel('Jumlah Dosen')
7 plt.grid(True)
8
9 # tampilkan visualisasi
10 plt.tight_layout()
11 plt.show()

```

Line chart biasanya digunakan untuk menggambarkan tren atau perubahan seiring waktu atau dalam konteks tertentu. Dalam line chart diatas, sumbu x menunjukkan jenjang pendidikan, sumbu y menunjukkan jumlah dosen yang memiliki jenjang pendidikan tersebut.



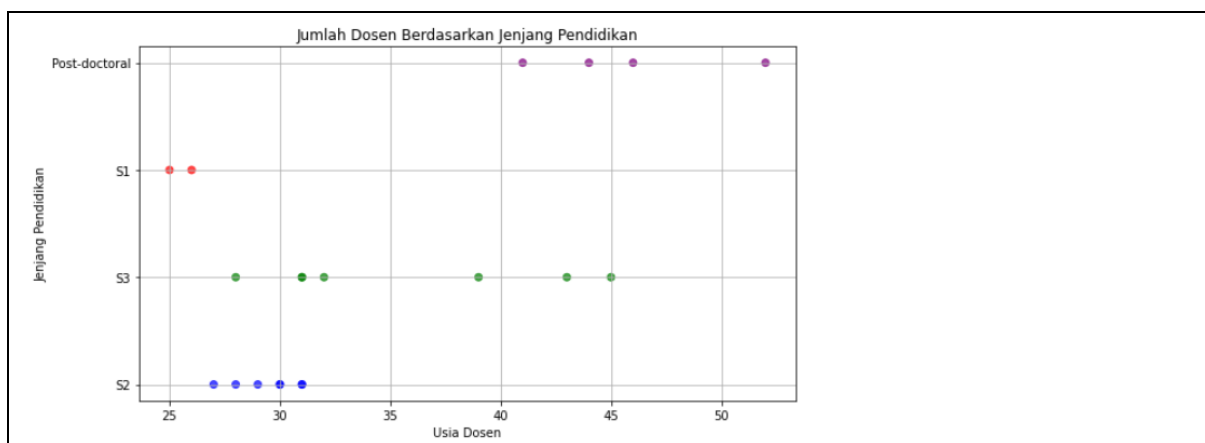
Dalam grafik, dapat dilihat bagaimana jumlah dosen berubah sepanjang urutane jenjang pendidikan. Garis biru yang menghubungkan titik-titik data menunjukkan bagaimana perkembangan jumlah dosen terjadi dari satu jenjang pendidikan ke jenjang pendidikan

berikutnya.

#### d) Scatter Plot

```
1 # Membuat scatter plot
2 plt.figure(figsize=(10, 6))
3 colors = {'S1': 'red', 'S2': 'blue', 'S3': 'green', 'Post-doctoral': 'purple'}
4 plt.scatter(df['Usia Sekarang'], df['Jenjang Pendidikan'], c=df['Jenjang Pendidikan'].map(colors), marker='o', alpha=0.7)
5 plt.title('Jumlah Dosen Berdasarkan Jenjang Pendidikan')
6 plt.xlabel('Usia Dosen')
7 plt.ylabel('Jenjang Pendidikan')
8 plt.grid(True)
9
10 # tampilkan visualisasi
11 plt.tight_layout()
12 plt.show()
```

Kode diatas menggambarkan scatter plot yang menggambarkan hubungan antara usia dosen dan jenjang pendidikan mereka. Scatter plot menggunakan titik-titik data untuk merepresentasikan setiap dosen dalam dataset.



Scatter plot diatas membantu kita untuk melihat sebaran dosen berdasarkan usia dan jenjang pendidikan mereka. Warna titik-titik menunjukkan jenjang pendidikan sedangkan posisi horizontal titik-titik menunjukkan usia dosen. Ini memberikan gambar visual tentang sebaran dan korelasi antara usia dan pendidikan dosen.

#### Kesimpulan

- 1) Bar chart menunjukkan distribusi jenjang pendidikan dosen. Mayoritas dosen memiliki jenjang pendidikan S2, diikuti S3, dan post-doctoral paling sedikit.
- 2) Pie chart menggambarkan proporsi dosen dalam setiap jenjang,. S2 mendominasi jumlah dosen yang signifikan.
- 3) Line chart menampilkan tren perubahan jumlah dosen sepanjang urutane jenjang pendidikan.
- 4) Scatter plot usia dosen vs jenjang pendidikan menggambarkan hubungan antara usia dosen dan jenjang pendidikan mereka.

Jadi, kombinasi visualisasi memberikan pemahaman yang lebih baik dalam profil dosen, distribusi jenjang, perubahan seiring waktu, proporsi relative, dan hubungan antara usia dan pendidikan dosen.

## 2. Daftar Pasien Diabetes Berdasarkan Faktor Gula Darah, Berat Badan, dan Tinggi Badan

Catatan: Tentukan terlebih dahulu apakah berat badan dikategorikan kurang (underweight), normal, berat badan berlebih (overweight) atau berat badan sangat berlebih (obesitas) berdasarkan parameter yang ada

No.	Tinggi Badan (cm)	Berat Badan (kg)	Gula Darah (Puasa)
1.	165	50	4.9
2.	150	95	5.7
3.	154	52	4.5
4.	149	43	8.14
5.	166	96	8.05
6.	156	67	4.2
7.	148	42	8.14
8.	154	64	4.89
9.	163	74	5.08
10.	152	98	5.6
11.	166	51	4.7
12.	151	96	5.8
13.	150	48	4.3
14.	146	40	8.19
15.	163	93	8.08
16.	151	61	4.3
17.	146	40	8.10
18.	156	66	4.92
19.	160	71	5.05
20.	147	95	5.3

Import library pandas dan input data menjadi DataFrame



```

1 import pandas as pd
2
3 # Data pasien
4 data = {
5     'Tinggi Badan (cm)': [165, 150, 154, 149, 166, 166, 156, 148, 154, 163, 152, 166, 151, 150, 146, 163, 151, 146,
6     'Berat Badan (kg)': [50, 95, 52, 43, 96, 67, 42, 64, 74, 98, 51, 96, 48, 40, 93, 61, 40, 66, 71, 95],
7     'Gula Darah (Puasa)': [4.9, 5.7, 4.5, 8.14, 8.05, 4.2, 8.14, 4.89, 5.08, 5.6, 4.7, 5.8, 4.3, 8.19, 8.08, 4.
8 }
9
10 # Membuat dataframe
11 df = pd.DataFrame(data)
12
13 # Menghitung BMI
14 df['BMI'] = df['Berat Badan (kg)'] / ((df['Tinggi Badan (cm)'] / 100) ** 2)
15
16 # Kategorikan berat badan dan analisis gula darah
17 def kategorikan_dan_analisis(row):
18     bmi = row['BMI']
19     gula_darah = row['Gula Darah (Puasa)']
20
21     if bmi < 18.5:
22         berat_badan = 'Kurang (Underweight)'
23     elif 18.5 <= bmi < 24.9:
24         berat_badan = 'Normal'
25     elif 25 <= bmi < 29.9:
26         berat_badan = 'Berat Badan Berlebih (Overweight)'
27     else:
28         berat_badan = 'Obesitas'
29
30     if gula_darah >= 5.6:
31         analisis_gula = 'Diabetes'
32     elif 5.6 > gula_darah >= 4.0:
33         analisis_gula = 'Prediabetes'
34     else:
35         analisis_gula = 'Normal Gula Darah'
36
37     return pd.Series([berat_badan, analisis_gula], index=['Kategori Berat Badan', 'Analisis Gula Darah'])
38
39 df[['Kategori Berat Badan', 'Analisis Gula Darah']] = df.apply(kategorikan_dan_analisis, axis=1)
40
41 # Menampilkan hasil analisis
42 (df[['Tinggi Badan (cm)', 'Berat Badan (kg)', 'Gula Darah (Puasa)', 'BMI', 'Kategori Berat Badan', 'Analisis Gula Darah']])

```

Kode diatas membuat DataFrame dari data yang mencakup tinggi badan, berat badan, dan gula darah puasa. Selanjutnya, kode menghitung Indeks Massa Tubuh (BMI) untuk setiap pasien dan mengkategorikan berat badan berdasarkan BMI. Hasil analisis akan menjadi DataFrame seperti berikut:

	Tinggi Badan (cm)	Berat Badan (kg)	Gula Darah (Puasa)	BMI	Kategori Berat Badan	Analisis Gula Darah
0	165	50	4.90	18.365473	Kurang (Underweight)	Prediabetes
1	150	95	5.70	24.222222	Obesitas	Diabetes
2	154	52	4.50	21.926126	Normal	Prediabetes
3	149	43	8.14	19.368497	Normal	Diabetes
4	166	96	8.05	34.838148	Obesitas	Diabetes
5	156	67	4.20	27.531229	Berat Badan Berlebih (Overweight)	Prediabetes
6	148	42	8.14	19.174580	Normal	Diabetes
7	154	64	4.89	26.986001	Berat Badan Berlebih (Overweight)	Prediabetes
8	163	74	5.08	27.852008	Berat Badan Berlebih (Overweight)	Prediabetes
9	152	98	5.60	42.416898	Obesitas	Diabetes
10	166	51	4.70	18.507766	Normal	Prediabetes
11	151	96	5.80	42.103417	Obesitas	Diabetes
12	150	48	4.30	21.333333	Normal	Prediabetes
13	146	40	8.19	18.765247	Normal	Diabetes
14	163	93	8.08	35.003199	Obesitas	Diabetes
15	151	61	4.30	26.753213	Berat Badan Berlebih (Overweight)	Prediabetes
16	146	40	8.10	18.765247	Normal	Diabetes
17	156	66	4.92	27.120316	Berat Badan Berlebih (Overweight)	Prediabetes
18	160	71	5.05	27.734375	Berat Badan Berlebih (Overweight)	Prediabetes
19	147	95	5.30	43.963163	Obesitas	Prediabetes

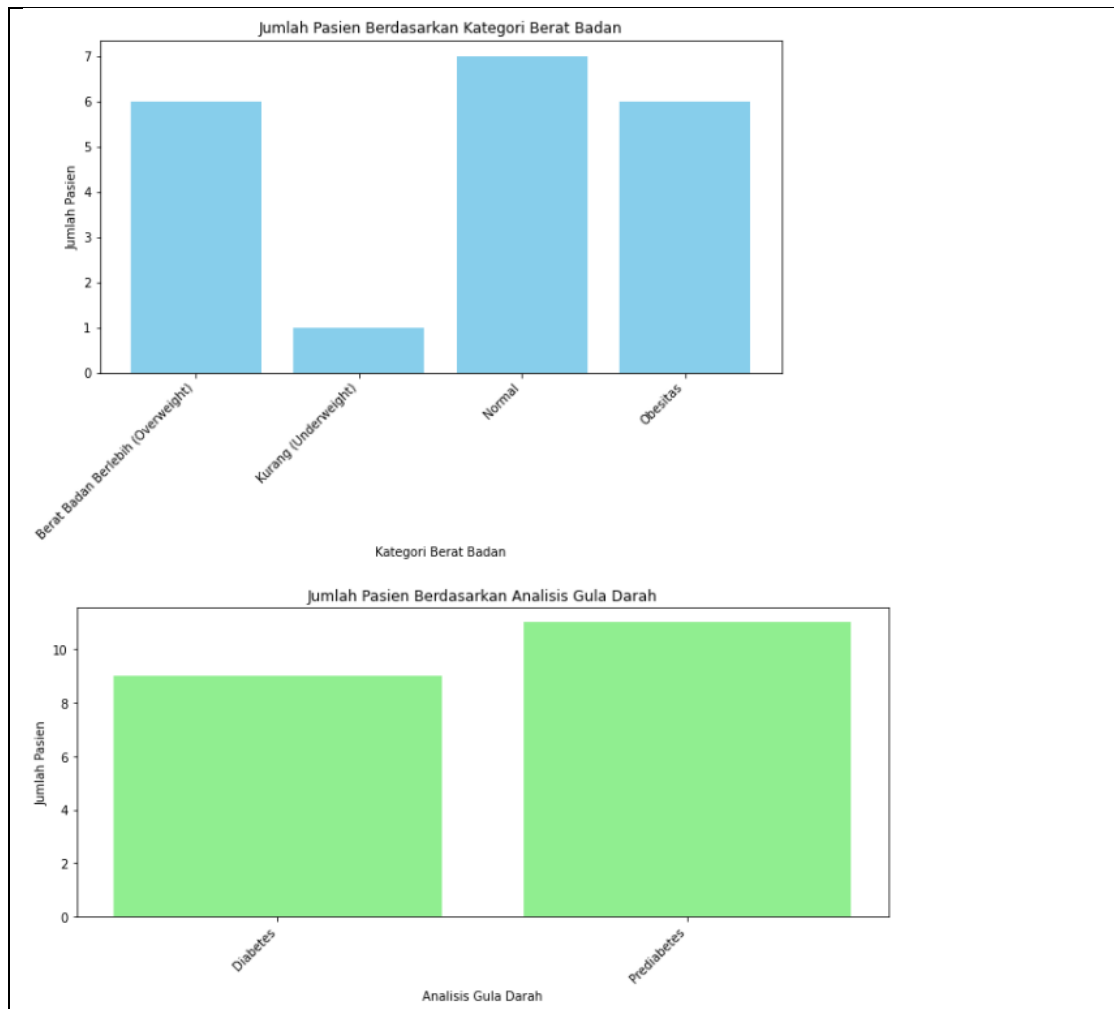
DataFrame terbaru mencakup tinggi badan, berat badan, gula darah, BMI, Kategori

berat badan, dan analisis gula darah.

Selanjutnya dapat dilakukan visualisasi menggunakan bar chart.

```
1 # Menghitung jumlah setiap kategori berat badan
2 jumlah_berat_badan = df['Kategori Berat Badan'].value_counts().sort_index()
3
4 # Menghitung jumlah setiap analisis gula darah
5 jumlah_gula_darah = df['Analisis Gula Darah'].value_counts().sort_index()
6
7 # visualisasi berat badan
8 plt.figure(figsize=(10, 5))
9 plt.bar(jumlah_berat_badan.index, jumlah_berat_badan, color='skyblue')
10 plt.title('Jumlah Pasien Berdasarkan Kategori Berat Badan')
11 plt.xlabel('Kategori Berat Badan')
12 plt.ylabel('Jumlah Pasien')
13 plt.xticks(rotation=45, ha='right')
14
15 # visualisasi gula darah
16 plt.figure(figsize=(10, 5))
17 plt.bar(jumlah_gula_darah.index, jumlah_gula_darah, color='lightgreen')
18 plt.title('Jumlah Pasien Berdasarkan Analisis Gula Darah')
19 plt.xlabel('Analisis Gula Darah')
20 plt.ylabel('Jumlah Pasien')
21 plt.xticks(rotation=45, ha='right')
22
23 plt.tight_layout()
24 plt.show()
```

Kode diatas melakukan perhitungan dan visualisasi terhadap data pasien, terutama berkaitan dengan berat badan dan analisis gula darah. Dua bar chart dibuat untuk memvisualisasikan hasil perhitungan tersebut. Bar chart pertama menunjukkan jumlah pasien dalam setiap kategori dan bar chart kedua menampilkan jumlah pasien berdasarkan hasil analisis gula darah. Berikut visualisasi bar chart:



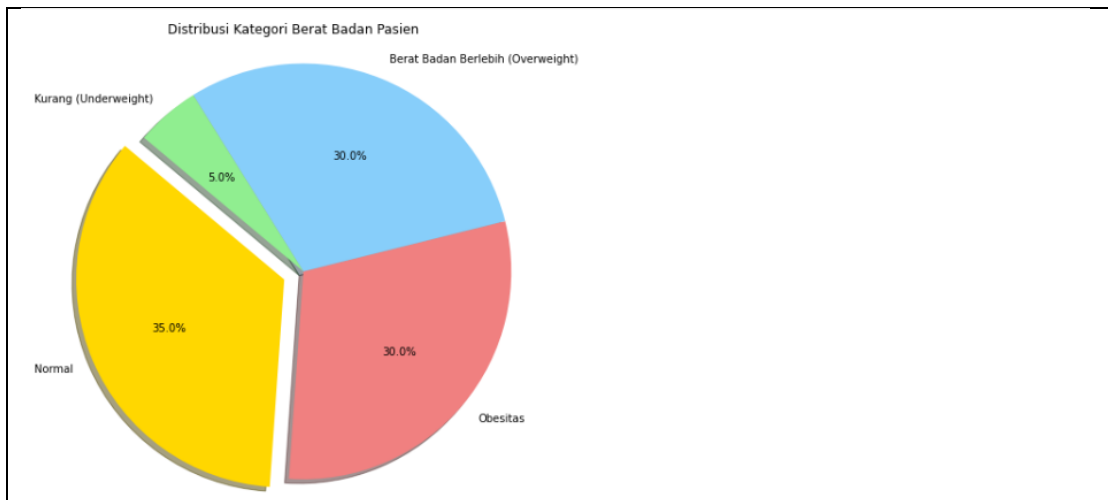
Selain bar chart, kita dapat juga menggunakan pie chart dalam visualisasi distribusi kategori berat badan (BMI).

```

1 # Hitung jumlah pasien dalam setiap kategori berat badan
2 kategori_count = df['Kategori Berat Badan'].value_counts()
3
4 # Siapkan data untuk pie chart
5 labels = kategori_count.index
6 sizes = kategori_count.values
7 colors = ['gold', 'lightcoral', 'lightskyblue', 'lightgreen']
8 explode = (0.1, 0, 0, 0)
9
10 # Membuat pie chart
11 plt.figure(figsize=(8, 8))
12 plt.pie(sizes, explode=explode, labels=labels, colors=colors, autopct='%1.1f%%', shadow=True, startangle=140)
13 plt.title('Distribusi Kategori Berat Badan Pasien')
14
15 # Menampilkan pie chart
16 plt.axis('equal')
17 plt.show()

```

Kode visualisasi diatas menampilkan pie chart yang mudah dipahami presentasinya.



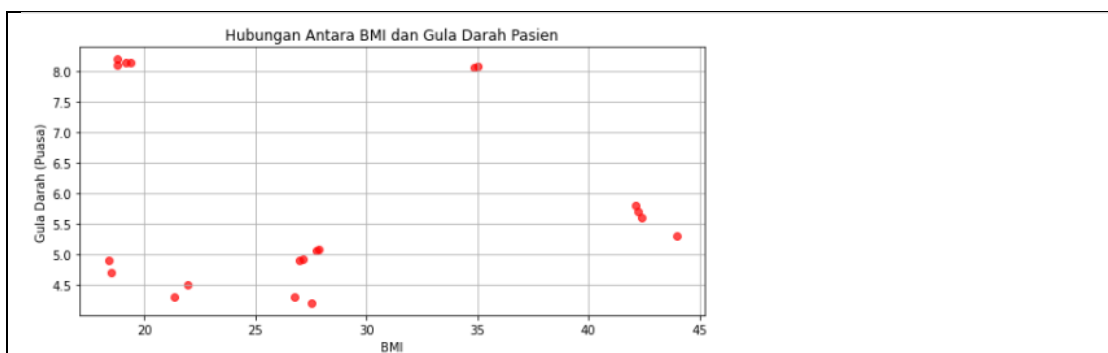
Scatter plot juga dapat menampilkan hubungan antara BMI dan Gula Darah Pasien.

```

1 # Membuat scatter plot
2 plt.figure(figsize=(10, 6))
3 plt.scatter(df['BMI'], df['Gula Darah (Puasa)'], c='red', marker='o', alpha=0.7)
4 plt.title('Hubungan Antara BMI dan Gula Darah Pasien')
5 plt.xlabel('BMI')
6 plt.ylabel('Gula Darah (Puasa)')
7 plt.grid(True)
8
9 # Menampilkan visualisasi
10 plt.tight_layout()
11 plt.show()

```

Scatter plot diatas menggambarkan hubungan BMI dan kadar gula darah (puasa) pada sekelompok pasien.



Melalui visualisasi ini, kita dapat melihat sebaran titik-titik data yang mewakili pasien. Posisi setiap titik dapat memberikan wawasan tentang hubungan antara BMI dan kadar gula darah pada pasien. Jika terdapat pola atau tren tertentu, scatter plot ini dapat membantu dalam mendeteksi korelasi atau hubungan antara dua variabel tersebut. Selain itu, warna merah yang digunakan untuk titik-titik data membantu membedakan titik-titik pada plot.

Jadi, pilihan visualisasi tergantung pada tujuan analisis dan informasi yang ingin anda sampaikan. Bar chart cocok untuk membandingkan kategor atau kelompok berbeda dalam BMI. Pie chart tidak begitu cocok untuk BMI, karena pie chart lebih cocok menampilkan presentase dari beberapa kategori, dalam BMI tidak ada kategori yang mencolok seperti dalam data kualitatif, dan scatter plot sangat cocok digunakan jika ingin memahami hubungan antara dua variable, seperti BMI dan kadar gula darah.

### 3. Daftar Kasus Pasien Terkonfirmasi Covid-19 Harian

Catatan: Hitung terlebih dahulu pertumbuhan kasus Covid-19 harian.

No.	Tanggal	Jumlah Kasus Terkonfirmasi
1.	02-Mar-20	2
2.	03-Mar-20	0
3.	04-Mar-20	0
4.	05-Mar-20	0
5.	06-Mar-20	2
6.	07-Mar-20	0
7.	08-Mar-20	2
8.	09-Mar-20	13
9.	10-Mar-20	8
10.	11-Mar-20	7
11.	12-Mar-20	0
12.	13-Mar-20	35
13.	14-Mar-20	27
14.	15-Mar-20	21
15.	16-Mar-20	17
16.	17-Mar-20	38
17.	18-Mar-20	55
18.	19-Mar-20	82
19.	20-Mar-20	60
20.	21-Mar-20	81
21.	22-Mar-20	64
22.	23-Mar-20	65
23.	24-Mar-20	106
24.	25-Mar-20	105
25.	26-Mar-20	103
26.	27-Mar-20	153
27.	28-Mar-20	109
28.	29-Mar-20	130
29.	30-Mar-20	129
30.	31-Mar-20	114

Input data diatas menjadi DataFrame terlebih dahulu.

```
1 import pandas as pd
2
3 # Data jumlah kasus terkonfirmasi COVID-19 harian
4 data = {
5     'Tanggal': [
6         '02-Mar-20', '03-Mar-20', '04-Mar-20', '05-Mar-20', '06-Mar-20', '07-Mar-20',
7         '08-Mar-20', '09-Mar-20', '10-Mar-20', '11-Mar-20', '12-Mar-20', '13-Mar-20',
8         '14-Mar-20', '15-Mar-20', '16-Mar-20', '17-Mar-20', '18-Mar-20', '19-Mar-20',
9         '20-Mar-20', '21-Mar-20', '22-Mar-20', '23-Mar-20', '24-Mar-20', '25-Mar-20',
10        '26-Mar-20', '27-Mar-20', '28-Mar-20', '29-Mar-20', '30-Mar-20', '31-Mar-20'
11    ],
12     'Jumlah Kasus Terkonfirmasi': [
13         2, 0, 0, 0, 2, 0, 2, 13, 8, 7, 0, 35, 27, 21, 17, 38, 55, 82, 60, 81, 64, 65, 106,
14         105, 103, 153, 109, 130, 129, 114
15    ]
16 }
17
18 # Membuat DataFrame
19 df = pd.DataFrame(data)
```

Setelah menjadi dataframe, kita memerlukan penyesuaian kolom dengan menambahkan satu atribut yaitu pertumbuhan harian untuk keperluan visualisasi nantinya.

```
1 # Menghitung pertumbuhan harian
2 df['Pertumbuhan Harian'] = df['Jumlah Kasus Terkonfirmasi'].diff().fillna(0)
3
4 # Menampilkan hasil
5 (df[['Tanggal', 'Jumlah Kasus Terkonfirmasi', 'Pertumbuhan Harian']])
```

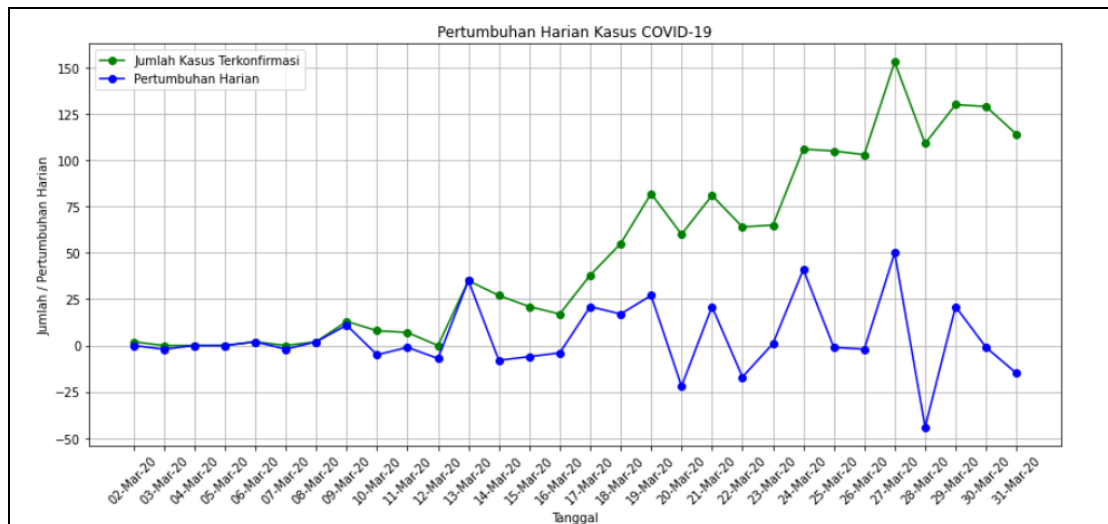
Menjadi DataFrame baru:

	Tanggal	Jumlah Kasus Terkonfirmasi	Pertumbuhan Harian
0	02-Mar-20	2	0.0
1	03-Mar-20	0	-2.0
2	04-Mar-20	0	0.0
3	05-Mar-20	0	0.0
4	06-Mar-20	2	2.0
5	07-Mar-20	0	-2.0
6	08-Mar-20	2	2.0
7	09-Mar-20	13	11.0
8	10-Mar-20	8	-5.0
9	11-Mar-20	7	-1.0
10	12-Mar-20	0	-7.0
11	13-Mar-20	35	35.0
12	14-Mar-20	27	-8.0
13	15-Mar-20	21	-6.0
14	16-Mar-20	17	-4.0
15	17-Mar-20	38	21.0
16	18-Mar-20	55	17.0
17	19-Mar-20	82	27.0
18	20-Mar-20	60	-22.0
19	21-Mar-20	81	21.0
20	22-Mar-20	64	-17.0
21	23-Mar-20	65	1.0
22	24-Mar-20	106	41.0
23	25-Mar-20	105	-1.0
24	26-Mar-20	103	-2.0
25	27-Mar-20	153	50.0
26	28-Mar-20	109	-44.0
27	29-Mar-20	130	21.0
28	30-Mar-20	129	-1.0
29	31-Mar-20	114	-15.0

Dari DataFrame terbaru, dapat divisualisasikan menjadi diagram garis (line chart) untuk melihat pertumbuhan harian kasus covid-19.

```
1 # Membuat line chart
2 plt.figure(figsize=(12, 6))
3
4 # Menampilkan jumlah kasus terkonfirmasi
5 plt.plot(df['Tanggal'], df['Jumlah Kasus Terkonfirmasi'], marker='o', linestyle='-', color='green', markersize=6)
6
7 # Menampilkan pertumbuhan harian
8 plt.plot(df['Tanggal'], df['Pertumbuhan Harian'], marker='o', linestyle='-', color='blue', markersize=6, label='Pertumbuhan Harian')
9
10 plt.title('Pertumbuhan Harian Kasus COVID-19')
11 plt.xlabel('Tanggal')
12 plt.ylabel('Jumlah / Pertumbuhan Harian')
13 plt.xticks(rotation=45)
14 plt.grid(True)
15 plt.legend()
16
17 # Menampilkan line chart
18 plt.tight_layout()
19 plt.show()
```

Kode visualisasi line chart diatas akan menggambarkan dua set data terkait kasus Covid-19.



Garis hijau menggambarkan jumlah kasus terkonfirmasi per hari atau setiap tanggal, sedangkan garis biru mewakili pertumbuhan harian kasus. Dengan visualisasi ini, kita dapat memahami perkembangan Covid-19 dengan melihat grafik naik turunnya dengan lebih jelas.

Jadi, grafik diatas membantu dalam melihat apakah ada peningkatan atau penurunan signifikan dalam jumlah kasus harian, serta memberikan gambaran umum tentang perkembangan pandemi.

### Kesimpulan:

Tiga dataset berbeda telah dianalisis, dataset pertama tentang profil dosen dapat dilakukan visualisasi berbagai jenis seperti bar chart, pie chart, line chart, dan scatter plot, sehingga kita dapat memahami profil dosen dalam hal jenjang pendidikan, usia, serta pendidikan mereka.

Dataset pasien Kesehatan digunakan untuk menganalisis kategori berat badan, analisis gula darah, dan hubungan antara BMI dan kadar gula darah. Melalui bar chart dan scatter plot, kita dapat memahami distribusi kategori berat badan dan analisis gula darah.

Dataset covid-19 cocok divisualisasikan menggunakan line chart, karena untuk pertumbuhan kasus harian dapat terlihat dan mudah dipahami visualnya, grafiknya sangat memungkinkan untuk memantau fluktuasi sehari-hari.