

LAPORAN UAS
PROYEK PEMROGRAMAN BERORIENTASI OBJEK



Disusun Oleh :

Nama Kelompok

- | | |
|----------------------|-------------|
| 1. Novia Al Qodri | (G1F022015) |
| 2. Suryaningsih | (G1F022049) |
| 3. F Yoshua Habeahan | (G1F022025) |

Asisten Dosen :

Rendi Julian Saputra	(G1A019066)
----------------------	-------------

Dosen Pengampu :

1. Ferzha Putra Utama, S.T.,M.Eng
2. Arie Vatesia, S.T.,M.TI,Ph.D.

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS BENGKULU
T.A 2023/202

BAB I

LANDASAN TEORI

1.1 Pemrograman Berorientasi Objek

Pemrograman Berorientasi Objek (PBO) atau *Object Oriented Programming* (OOP) adalah suatu metode pemrograman yang berorientasi kepada objek. Tujuan dari PBO/OOP diciptakan adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari. Jadi setiap bagian dari suatu permasalahan adalah objek, sedangkan objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil lagi. *Object oriented programming* atau OOP adalah sebuah konsep programming yang berorientasi pada data atau objek untuk mengatur desain program. Program yang sudah ada merupakan gabungan dari objek-objek yang saling berhubungan dan disusun menjadi satu class.

Dengan begitu, OOP akan membantu *Developer* menyelesaikan sejumlah masalah program yang cukup rumit. Definisi lain dari OOP yaitu konsep yang mengidentifikasi object class terkait metode yang digunakan dalam pemrograman. Konsep OOP menawarkan potensi development menuju tingkat abstraksi lebih tinggi. Oleh karena itu, OOP menjadi model pemrograman paling populer dan sering digunakan oleh para pengembang, baik pemula maupun yang sudah berpengalaman. Selain itu, OOP sendiri sudah mendukung berbagai bahasa pemrograman seperti *Python*, Java, JavaScript, dan C++. Penggunaan kelas dalam OOP dapat menentukan atribut instance objek, misalnya warna, bentuk, dan sebagainya. OOP memungkinkan pembuatan program secara ringkas, pasalnya *Developer* atau programmer bisa lebih fokus memanipulasi objek daripada function atau logiS.

Berikut ini adalah beberapa kelebihan OOP:

1. Skalabilitas Prinsip-prinsip dalam konsep OOP akan mempermudah kebutuhan program dengan skala yang lebih luas dan rumit. Hal ini memungkinkan *Developer* menambahkan beberapa objek, class, dan function secara mudah ketika mengembangkan suatu program. Prosedur pembuatan kode juga dapat dijaga dan dipelihara dengan cara meningkatkan keamanan programming dalam OOP.
2. Pengembangan paralel Kelebihan OOP selanjutnya yaitu parallel development atau pengembangan paralel. Ketika pengembangan program dilakukan secara tim, masing-masing programmer dapat bekerja sama dengan membuat class sendiri. Komponen atau class yang telah dibuat nantinya akan digabungkan menjadi satu kesatuan. Selain mempermudah pekerjaan, hal ini juga akan menghemat waktu pengembangan program.
3. Dapat digunakan kembali Berbagai class yang telah dibuat sebelumnya dapat digunakan kembali di masa mendatang. Keuntungan ini memudahkan pengembang membuat class

pada proyek lain yang sejenis. Jika class tidak sesuai, pengembang hanya perlu sedikit memodifikasinya sesuai kebutuhan.

Berikut ini adalah beberapa kekurangan OOP:

1. Kemungkinan duplikasi OOP memungkinkan pengembangan program baru dari program yang sudah ada sehingga mempermudah pekerjaan programmer. Meski begitu, kemudahan yang ditawarkan tersebut bisa menjadi sebuah bumerang karena rentan duplikasi. Hal ini disebabkan oleh fleksibilitas class dalam program yang amat luas.
2. Memerlukan data management yang ketat Meski memiliki keunggulan dari segi skalabilitas, namun OOP berpotensi lepas kendali jika terjadi kondisi kelebihan kapasitas. Kondisi ini biasanya terjadi karena munculnya kode-kode baru yang berasal dari kode dengan fungsi kurang baik. Oleh sebab itu, diperlukan manajemen data cukup ketat terhadap kode-kode yang telah dibuat dalam jumlah besar.
3. Kurang ramah spesifikasi perangkat Dalam penggunaannya, OOP bisa dibilang kurang efisien karena tidak ramah spesifikasi. OOP memakan banyak beban processor komputer sehingga penggunaan tidak disarankan apabila terdapat batasan teknis pada perangkat. Sebaiknya gunakan perangkat komputer terbaru jika ingin melakukan pengembangan program dengan OOP.

OOP memiliki 4 prinsip, diantaranya yaitu:

1. *Encapsulation* atau enkapsulasi
2. *Abstraction* atau abstraksi
3. *Inheritance* atau pewarisan
4. *Polymorphism*

1.2 Python

Python adalah bahasa pemrograman yang banyak digunakan dalam aplikasi web, pengembangan perangkat lunak, ilmu data, dan machine learning (ML). *Developer* menggunakan *Python* karena efisien dan mudah dipelajari serta dapat dijalankan di berbagai platform. Perangkat lunak *Python* dapat diunduh secara gratis, terintegrasi baik dengan semua tipe sistem, dan meningkatkan kecepatan pengembangan.

Python biasa dipakai dalam pengembangan situs dan perangkat lunak, membuat analisis data, *visualisasi* data dan otomatisasi tugas. Karena sifatnya yang relatif mudah dipelajari, bahasa pemrograman ini digunakan secara luas oleh non-programmer seperti ilmuwan dan akuntan untuk melakukan tugas harian mereka. Misalnya, dalam mengatur keuangan.

Manfaat *Python* mencakup:

1. *Developer* dapat dengan mudah membaca dan memahami program *Python* karena memiliki sintaksis dasar seperti dalam bahasa Inggris.

2. *Python* membuat *Developer* lebih produktif karena mereka dapat menulis program *Python* menggunakan baris kode yang lebih sedikit dibandingkan dengan bahasa lainnya.
3. *Python* memiliki pustaka standar besar berisi kode yang dapat digunakan kembali untuk hampir semua tugas. Sehingga, *Developer* tidak perlu menulis kode dari awal.
4. *Developer* dapat dengan mudah menggunakan *Python* dengan bahasa pemrograman populer lain seperti Java, C, dan C++.
5. Komunitas aktif *Python* memiliki jutaan *Developer* pendukung di seluruh dunia. Jika Anda mengalami kesulitan, Anda bisa mendapatkan dukungan cepat dari komunitas.
6. Tersedia banyak sumber daya bermanfaat di internet jika Anda ingin mempelajari *Python*. Contohnya, Anda dapat dengan mudah menemukan video, tutorial, dokumentasi, dan panduan *Developer*.
7. *Python* portabel untuk beragam sistem operasi komputer seperti *Windows*, *macOS*, *Linux*, dan *Unix*.

1.3 Fungsi *Python*

Python dapat digunakan untuk membangun dan mengembangkan aplikasi berbasis website, mobile, maupun desktop. Dalam proses pembangunan dan pengembangan ini, Anda membutuhkan bantuan Framework. *Python* digunakan untuk mengolah hal-hal yang berkaitan dengan data (*data science*). *Python* adalah bahasa pemrograman yang akan memudahkan seorang data scientist dalam mengekstraksi, membersihkan, hingga memvisualisasi data dengan mudah. Data scientist menjadi profesi yang banyak diminati karena semakin banyak perusahaan yang mencari ahli dalam pengolahan data. *Python* dapat dipilih karena lebih fleksibel jika dibandingkan dengan bahasa pemrograman lain. *Library* khusus di *Python* yang digunakan untuk data science, antara lain Numpy, Pandas, dan Matplotlib.

Data analysis masih memiliki keterkaitan dengan data science yang telah dibahas sebelumnya. *Library* yang digunakan pun masih sama. Perbedaannya adalah pada data analysis, *Python* digunakan untuk menganalisa data yang dihasilkan agar menjadi informasi yang lebih bermanfaat. *Python* juga digunakan untuk keperluan machine learning. *Python* dipilih untuk dapat membuat sistem machine learning *Python* adalah bahasa pemrograman yang dinilai sederhana, mudah dimengerti, fleksibel, efisien, dan multi-implementasi. *Library Python* yang digunakan untuk *machine learning*, kebanyakan bisa digunakan untuk mengembangkan *artificial intelligence*.

1.4 Sejarah perilisian *Python*

1. Guido Van Rossum memublikasikan versi pertama kode *Python* (versi 0.9.0) di tahun 1991. Versi ini sudah memiliki fitur yang bagus seperti beberapa tipe data dan fungsi untuk menangani kesalahan.

2. *Python* 1.0 dirilis tahun 1994 dengan fungsi baru untuk memudahkan proses pencantuman data, seperti memetakan, memfilter, dan mengurangi.
3. *Python* 2.0 dirilis pada tanggal 16 Oktober 2000, dengan fitur baru yang berguna bagi pemrogram, seperti dukungan untuk karakter *Unicode* dan *me-loop* daftar secara lebih singkat.
4. Pada 3 Desember 2008, *Python* 3.0 dirilis. Versi ini mencakup fitur seperti fungsi cetak dan lebih banyak dukungan untuk pembagian nomor dan penanganan masalah.

1.5 Visualisasi Data dan Pustaka Python

Visualisasi Data adalah penyajian data dalam format grafik. Ini membantu orang memahami pentingnya data dengan meringkas dan menyajikan sejumlah besar data dalam format yang sederhana dan mudah dipahami dan membantu mengkomunikasikan informasi dengan jelas dan efektif. Ini adalah disiplin mencoba memahami data dengan menempatkannya dalam konteks visual sehingga pola, tren, dan korelasi yang mungkin tidak terdeteksi dapat diekspos.

Python menawarkan beberapa pustaka grafik hebat yang dikemas dengan banyak fitur berbeda. Tidak masalah jika Anda ingin membuat *plot* interaktif, langsung, atau sangat disesuaikan, *Python* memiliki perpustakaan yang sangat baik untuk Anda. Untuk mendapatkan sedikit ikhtisar, berikut adalah beberapa pustaka *plot* yang populer:

- *Matplotlib* : level rendah, memberikan banyak kebebasan, *Matplotlib* adalah pustaka *plot* *Python* paling populer. Ini adalah perpustakaan tingkat rendah dengan antarmuka seperti *Matlab* yang menawarkan banyak kebebasan dengan biaya harus menulis banyak kode.
- *Visualisasi* *Panda*: antarmuka yang mudah digunakan, dibangun di atas *Matplotlib*, *Visualisasi* *Panda* membuatnya sangat mudah untuk membuat *plot* dari kerangka data dan seri *panda*. Ini juga memiliki API tingkat yang lebih tinggi daripada *Matplotlib* dan oleh karena itu kami membutuhkan lebih sedikit kode untuk hasil yang sama..
- *Seaborn*: antarmuka tingkat tinggi, gaya default yang bagus, *Seaborn* adalah pustaka *visualisasi* data *Python* berdasarkan *Matplotlib*. Ini menyediakan antarmuka tingkat tinggi untuk membuat grafik yang menarik.
- *ggplot*: berdasarkan *ggplot2* R, menggunakan Tata Bahasa Grafik, *Plot* Kotak adalah metode grafis untuk menampilkan ringkasan lima angka . Kita dapat membuat *plot* kotak menggunakan *sns.boxplot* metode *Seaborns* dan meneruskannya data serta nama kolom x dan y.
- *Plotly*: dapat membuat *plot interaktif*, representasi grafis dari data di mana nilai-nilai individual yang terkandung dalam matriks direpresentasikan sebagai warna. Peta panas sangat cocok untuk menjelajahi korelasi fitur dalam kumpulan data.

BAB II

SOAL DAN PEMBAHASAN

2.1 Soal

1. Menerapkan Penggunaan grafik dengan *Python*: Cara terbaik *Python* adalah *visualisasi*. Ini termasuk mempelajari berbagai tahapan pemrograman menggunakan pustaka *Python*

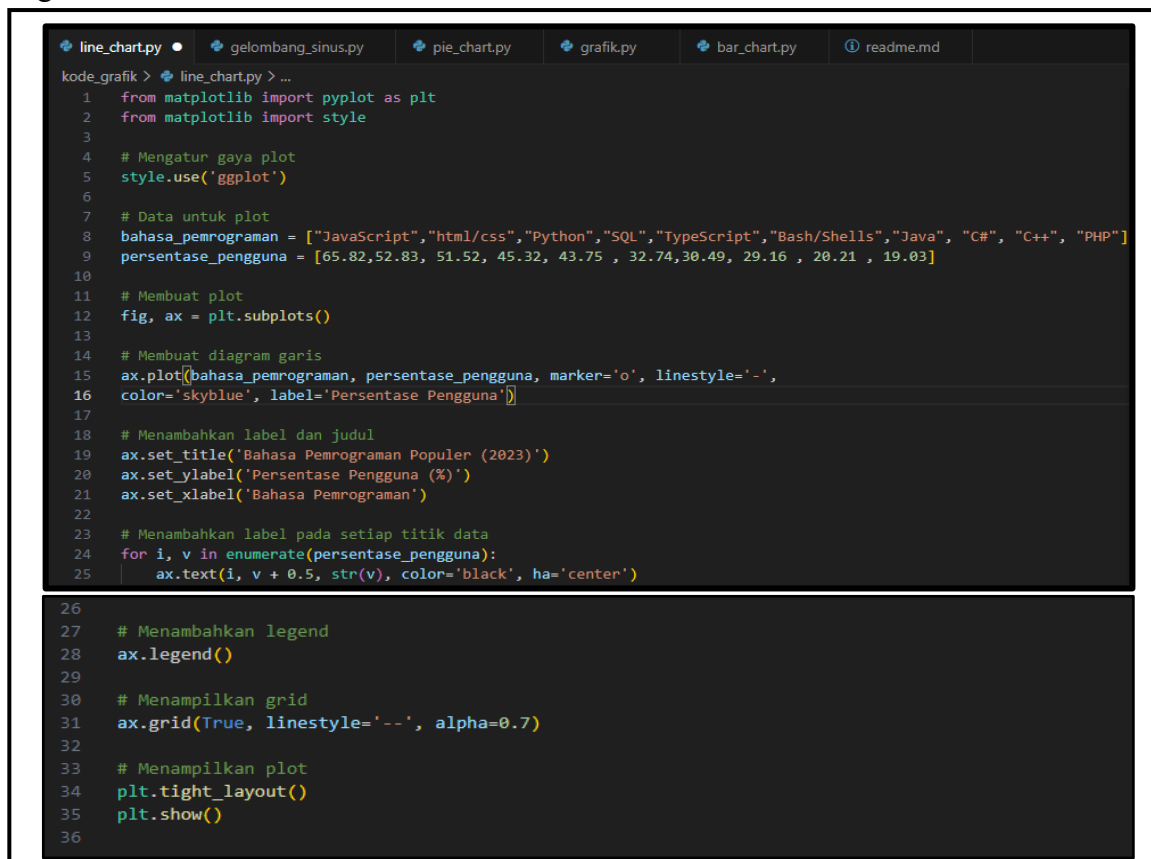
2.2 Pembahasan

Dalam menerapkan penggunaan grafik dengan *Python*, membuat *visualisasi* grafik ini kita dapat menggunakan berbagai tahapan pemrograman menggunakan pustaka/*Library Python*. Dalam Konteks ini kita menggunakan *Library/Pustakam Python Matplotlib*, disini kita menerapkan beberapa penggunaan grafik di *Python* seperti berikut :

➤ *Line chart* (Grafik Garis)

Line chart paling tepat digunakan untuk menunjukkan tren dari waktu ke waktu. Sumbu X biasanya mewakili periode waktu, sumbu Y menggambarkan nilai/kuantitas. Contohnya jumlah penjualan/ penggunaan dari pekan ke pekan selama satu tahun.

Grafik ini dapat memuat banyak titik data yang dapat diatur saling berdekatan sesuai kerapatan periode waktu. Karena visualnya yang simpel, bisa menggunakan banyak garis sekaligus dalam satu tampilan. Ini memudahkan penggambaran data tren dari beragam kategori.



```
line_chart.py • gelombang_sinus.py • pie_chart.py • grafik.py • bar_chart.py • readme.md
kode_grafik > line_chart.py > ...
1 from matplotlib import pyplot as plt
2 from matplotlib import style
3
4 # Mengatur gaya plot
5 style.use('ggplot')
6
7 # Data untuk plot
8 bahasa_pemrograman = ["JavaScript","html/css","Python","SQL","TypeScript","Bash/Shell","Java", "C#", "C++", "PHP"]
9 persentase_pengguna = [65.82,52.83, 51.52, 45.32, 43.75 , 32.74,30.49, 29.16 , 20.21 , 19.03]
10
11 # Membuat plot
12 fig, ax = plt.subplots()
13
14 # Membuat diagram garis
15 ax.plot(bahasa_pemrograman, persentase_pengguna, marker='o', linestyle='--',
16 color='skyblue', label='Persentase Pengguna')
17
18 # Menambahkan label dan judul
19 ax.set_title('Bahasa Pemrograman Populer (2023)')
20 ax.set_ylabel('Persentase Pengguna (%)')
21 ax.set_xlabel('Bahasa Pemrograman')
22
23 # Menambahkan label pada setiap titik data
24 for i, v in enumerate(persentase_pengguna):
25     ax.text(i, v + 0.5, str(v), color='black', ha='center')
26
27 # Menambahkan legend
28 ax.legend()
29
30 # Menampilkan grid
31 ax.grid(True, linestyle='--', alpha=0.7)
32
33 # Menampilkan plot
34 plt.tight_layout()
35 plt.show()
36
```

Gambar 2.1 Tampilan *Code Line chart*

- *Source Code*

```
from Matplotlib import pyplot as plt
from Matplotlib import style

# Mengatur gaya plot
style.use('ggplot')

# Data untuk plot
bahasa_pemrograman =
["JavaScript", "html/css", "Python", "SQL", "TypeScript", "Bash/Shells", "Java", "C#", "C++",
"PHP"]
persentase_pengguna = [65.82, 52.83, 51.52, 45.32, 43.75, 32.74, 30.49, 29.16, 20.21, 19.03]

# Membuat plot
fig, ax = plt.subplots()

# Membuat diagram garis
ax.plot(bahasa_pemrograman, persentase_pengguna, marker='o', linestyle='-', color='skyblue',
label='Persentase Pengguna')

# Menambahkan label dan judul
ax.set_title('Bahasa Pemrograman Populer (2023)')
ax.set_ylabel('Persentase Pengguna (%)')
ax.set_xlabel('Bahasa Pemrograman')

# Menambahkan label pada setiap titik data
for i, v in enumerate(persentase_pengguna):
    ax.text(i, v + 0.5, str(v), color='black', ha='center')

# Menambahkan legend
ax.legend()

# Menampilkan grid
ax.grid(True, linestyle='--', alpha=0.7)
```

Menampilkan *plot*

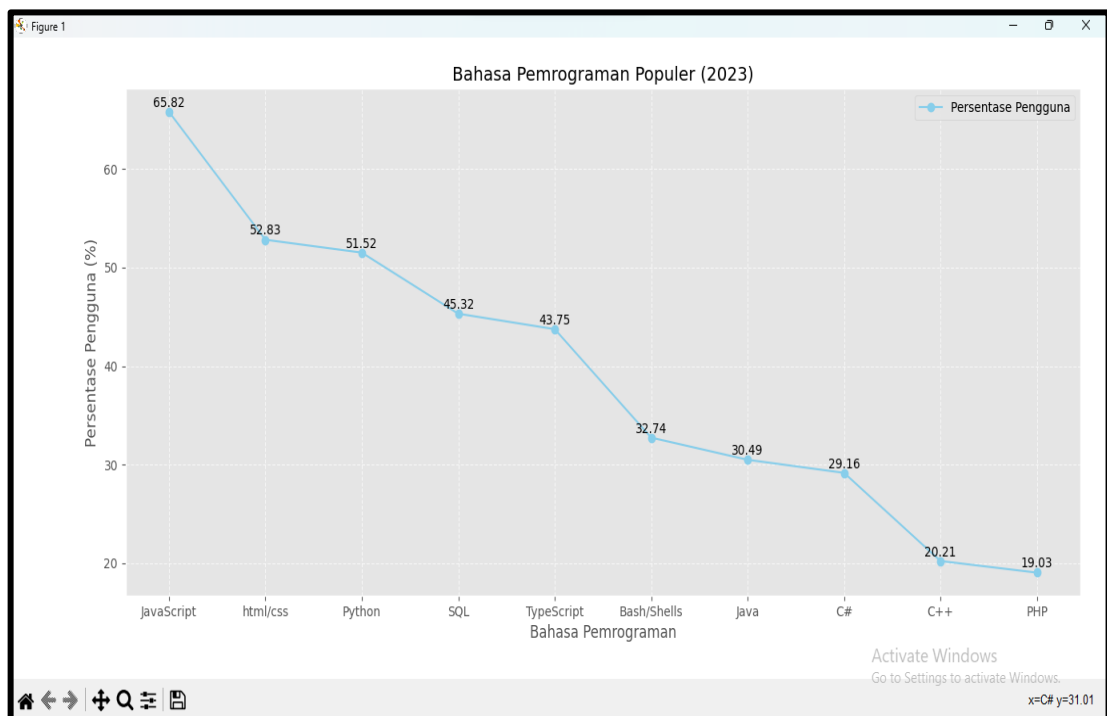
`plt.tight_layout()`

`plt.show()`

- Penjelasan :

Gambar 2.1 dan *Source Code* diatas adalah sebuah script dalam bahasa *Python* yang menggunakan pustaka *Matplotlib* untuk membuat diagram garis (line chart) yang menggambarkan persentase pengguna berbagai bahasa pemrograman yang sedang populer pada tahun 2023. Kode ini mengimpor dua modul dari pustaka *Matplotlib*, yaitu *pyplot* untuk membuat *plot* dan *style* untuk mengatur gaya *plot*. Kemudian kita mengatur gaya *plot* menggunakan gaya '*ggplot*' dari *Matplotlib*. Serta menyediakan data yang akan digunakan untuk membuat *plot*. Bahasa pemrograman adalah daftar nama bahasa pemrograman, dan persentase pengguna adalah persentase pengguna untuk masing-masing bahasa. Kemudian kita dapat Membuat diagram garis dengan menggunakan data yang telah disediakan, memberikan penanda (marker), gaya garis (linestyle), warna (color), dan label untuk legenda dan menambahkan judul dan label pada sumbu-x dan sumbu-y. Selanjutnya kita menambahkan label pada setiap titik data untuk menunjukkan persentase pengguna, Menambahkan legenda pada *plot*, Menampilkan grid pada *plot* dengan garis putus-putus dan yang terakhir menampilkan *plot* setelah konfigurasi selesai. Fungsi `tight_layout()` digunakan agar elemen-elemen *plot* tidak tumpang tindih. Fungsi `show()` menampilkan *plot* secara keseluruhan.

- Luaran



Gambar 2.2 Tampilan Luaran/Output *Line chart*

- Penjelasan :

Gambar 2.2 diatas merupakan tampilan atau *Output* dari source kode dan gambar 2.1 dimana didalamnya terdapat tampilan yang sebelumnya kita panggil. Yang dimana ada judul yaitu bahasa pemrograman populer (2023), 10 daftar nama yang dimasukan dan presentase penggunaanya atau berapa persen pengguna bahasa yang juga terdapat hasil peresentase dari bahasa pemograman yang populer pada tahun 2023 pada garis garis grafik tersebut.

➤ **Bar Chart (Grafik Batang)**

Bar Chart paling cocok untuk komparasi data dengan banyak kategori atau rangkaian data (*data series*). Untuk kemudahan membaca data, dapat mengurutkan kategori berdasarkan besar nilainya, misal dari nilai tertinggi hingga terendah. Lain halnya dengan data series, di mana data didistribusikan berdasarkan kategori berjenjang, misalnya populasi penduduk berdasarkan rentang usia atau tingkat pendidikan dan penggunaan bahasa pemograman.

```

kode_grafik > bar_chart.py > ...
1  from matplotlib import pyplot as plt
2  from matplotlib import style
3
4  # Mengatur gaya plot
5  style.use('ggplot')
6
7  # Data untuk plot
8  bahasa_pemrograman = ["JavaScript", "html/css", "Python", "SQL", "TypeScript", "Bash/Shell", "Java", "C#", "C++", "PHP"]
9  persentase_pengguna = [65.82, 52.83, 51.52, 45.32, 43.75, 32.74, 30.49, 29.16, 20.21, 19.03]
10
11 # Membuat plot
12 fig, ax = plt.subplots()
13
14 # Membuat diagram batang
15 ax.bar(bahasa_pemrograman, persentase_pengguna, align='center', color='skyblue')
16
17 # Menambahkan label dan judul
18 ax.set_title('Bahasa Pemrograman Populer (2023)')
19 ax.set_ylabel('Persentase Pengguna (%)')
20 ax.set_xlabel('Bahasa Pemrograman')
21
22 # Menambahkan label pada sumbu x
23 ax.set_xticks(bahasa_pemrograman)
24 ax.set_xticklabels(bahasa_pemrograman, rotation=45, ha='right')
25
26 # Menambahkan label pada setiap bar
27 for i, v in enumerate(persentase_pengguna):
28     ax.text(i, v + 0.5, str(v), color='black', ha='center')
29
30 # Menampilkan plot
31 plt.tight_layout() # Untuk memastikan label tidak tumpang tindih
32 plt.show()

```

Gambar 2.3 Tampilan *Code Bar Chart*

- *Source Code*

```

from Matplotlib import pyplot as plt
from Matplotlib import style
# Mengatur gaya plot

```

```

style.use('ggplot')
# Data untuk plot
bahasa_pemrograman =
["JavaScript","html/css","Python","SQL","TypeScript","Bash/Shells","Java", "C#", "C++",
"PHP"]
persentase_pengguna = [65.82,52.83, 51.52, 45.32, 43.75 , 32.74,30.49, 29.16 , 20.21 , 19.03]

# Membuat plot
fig, ax = plt.subplots()

# Membuat diagram batang
ax.bar(bahasa_pemrograman, persentase_pengguna, align='center', color='skyblue')

# Menambahkan label dan judul
ax.set_title('Bahasa Pemrograman Populer (2023)')
ax.set_ylabel('Persentase Pengguna (%)')
ax.set_xlabel('Bahasa Pemrograman')

# Menambahkan label pada sumbu x
ax.set_xticks(bahasa_pemrograman)
ax.set_xticklabels(bahasa_pemrograman, rotation=45, ha='right')

# Menambahkan label pada setiap bar
for i, v in enumerate(persentase_pengguna):
    ax.text(i, v + 0.5, str(v), color='black', ha='center')

# Menampilkan plot
plt.tight_layout() # Untuk memastikan label tidak tumpang tindih
plt.show()

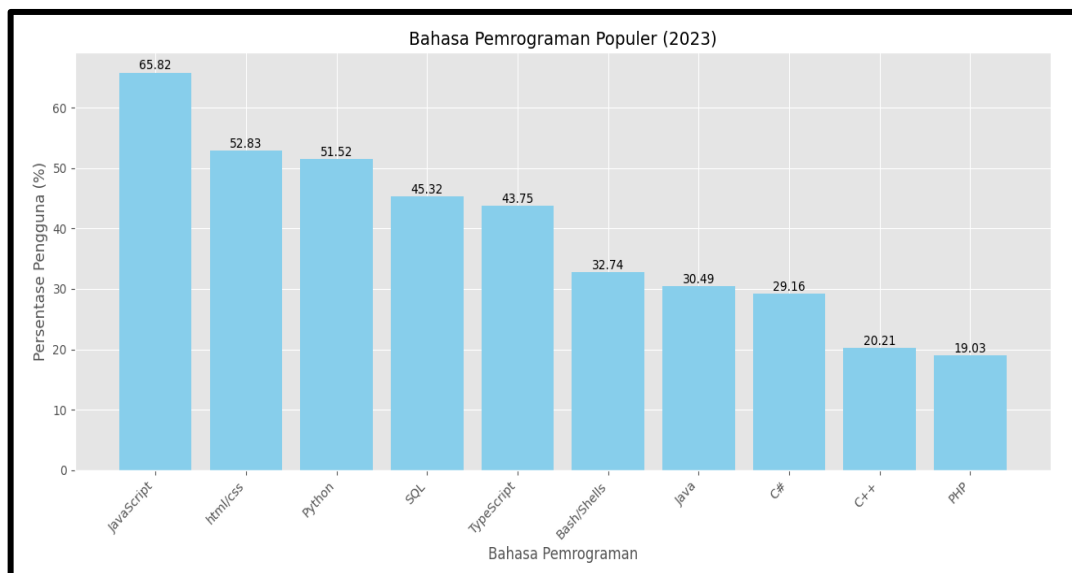
```

- Penjelasan :

Gambar 2.3 dan *Source Code* diatas adalah sebuah script dalam bahasa *Python* yang menggunakan pustaka *Matplotlib* untuk membuat diagram batang (*Bar Chart*) yang menggambarkan persentase pengguna berbagai bahasa pemrograman pada tahun 2023. Kode ini kurang lebih sama dengan kode sebelumnya dimana Kode ini mengimpor dua modul dari pustaka *Matplotlib*, yaitu *pyplot* untuk membuat *plot* dan *style* untuk mengatur gaya *plot* dan

mengatur gaya *plot* menggunakan gaya *ggplot* dari *Matplotlib*. Kemudian Menyediakan data yang akan digunakan untuk membuat *plot*. Bahasa pemrograman adalah daftar nama bahasa pemrograman, dan persentase pengguna adalah persentase pengguna untuk masing-masing bahasa. Kemudian Membuat objek gambar (*fig*) dan sumbu (*ax*) menggunakan fungsi *subplots()*. Setelahnya kita Menambahkan judul dan label pada sumbu-x dan sumbu-y dan Menambahkan label pada sumbu-x (bahasa pemrograman) dan mengatur rotasi label agar lebih mudah dibaca, Menambahkan label pada setiap batang untuk menunjukkan persentase pengguna dan kemudian yang terakhir Menampilkan *plot* setelah konfigurasi selesai. Fungsi *tight_layout()* digunakan agar elemen-elemen *plot* tidak tumpang tindih. Fungsi *show()* menampilkan *plot* secara keseluruhan.

- Luaran



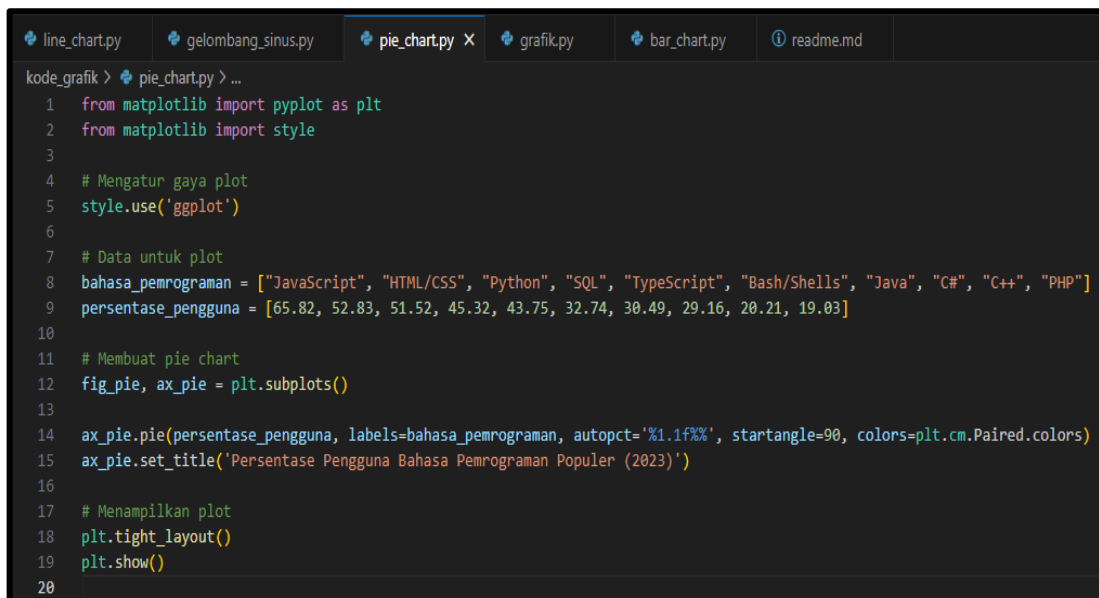
Gambar 2.4 Tampilan Luaran/Output Bar Chart

- Penjelasan :

Gambar 2.4 diatas merupakan tampilan atau *Output* dari source kode dan gambar 2.3 dimana didalamnya terdapat tampilan yang sebelumnya kita panggil. Yang dimana ada judul yaitu bahasa pemrograman populer (2023), 10 daftar nama yang dimasukan dan presentase penggunaanya atau berapa persen pengguna bahasa, yang juga terdapat hasil peresentase dari bahasa pemrograman yang populer pada tahun 2023 yang ditampilkan dalam bentuk batang batang grafik tersebut.

➤ *Pie Chart* (Grafik/ Diagram Lingkaran)

Pie Chart digunakan untuk menggambarkan komposisi antarbagian pada suatu kesatuan utuh. Bagian ini biasanya direpresentasikan dalam satuan persen sehingga jika seluruh bagian dijumlahkan, hasilnya sama dengan seratus persen. Jenis grafik ini akan mudah dipahami jika kategori yang ditampilkan tidak banyak, misal 5 bagian. Semakin banyak bagiannya, apalagi jika proposinya sangat kecil, akan kian sulit membacanya.



```
kode_grafik > pie_chart.py > ...
1  from matplotlib import pyplot as plt
2  from matplotlib import style
3
4  # Mengatur gaya plot
5  style.use('ggplot')
6
7  # Data untuk plot
8  bahasa_pemrograman = ["JavaScript", "HTML/CSS", "Python", "SQL", "TypeScript", "Bash/Shells", "Java", "C#", "C++", "PHP"]
9  persentase_pengguna = [65.82, 52.83, 51.52, 45.32, 43.75, 32.74, 30.49, 29.16, 20.21, 19.03]
10
11 # Membuat pie chart
12 fig_pie, ax_pie = plt.subplots()
13
14 ax_pie.pie(persentase_pengguna, labels=bahasa_pemrograman, autopct='%1.1f%%', startangle=90, colors=plt.cm.Paired.colors)
15 ax_pie.set_title('Persentase Pengguna Bahasa Pemrograman Populer (2023)')
16
17 # Menampilkan plot
18 plt.tight_layout()
19 plt.show()
20
```

Gambar 2.5 Tampilan Code Pie Chart

- Source Code

```
from Matplotlib import pyplot as plt
```

```
from Matplotlib import style
```

```
# Mengatur gaya plot
```

```
style.use('ggplot')
```

```
# Data untuk plot
```

```
bahasa_pemrograman = ["JavaScript", "HTML/CSS", "Python", "SQL", "TypeScript",  
"Bash/Shells", "Java", "C#", "C++", "PHP"]
```

```
persentase_pengguna = [65.82, 52.83, 51.52, 45.32, 43.75, 32.74, 30.49, 29.16, 20.21, 19.03]
```

```
# Membuat pie chart
```

```
fig_pie, ax_pie = plt.subplots()
```

```
ax_pie.pie(persentase_pengguna, labels=bahasa_pemrograman, autopct='%1.1f%%',  
startangle=90, colors=plt.cm.Paired.colors)
```

```
ax_pie.set_title('Persentase Pengguna Bahasa Pemrograman Populer (2023)')
```

```
# Menampilkan plot
```

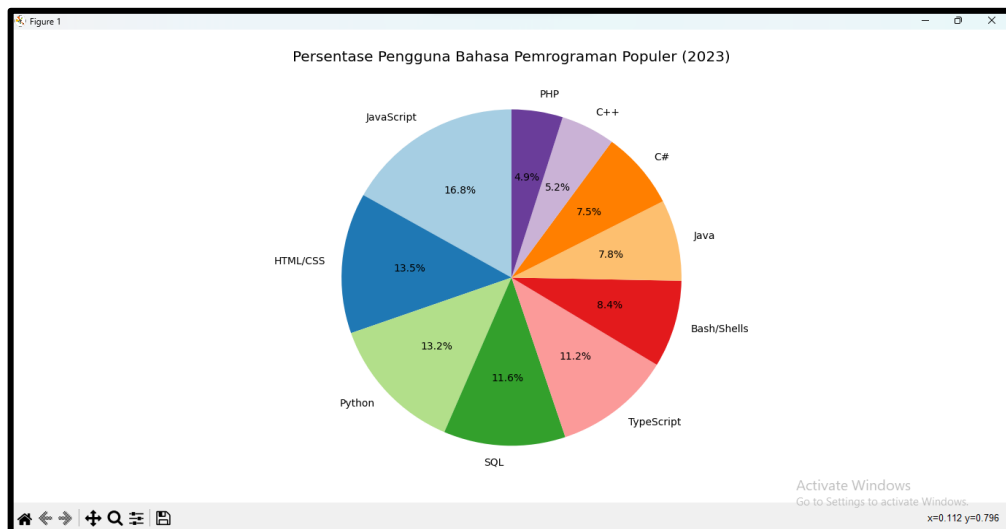
```
plt.tight_layout()
```

```
plt.show()
```

- Penjelasan :

Gambar 2.5 dan *Source Code* diatas adalah script dalam bahasa *Python* yang menggunakan pustaka *Matplotlib* untuk membuat diagram lingkaran (*pie chart*) yang menggambarkan persentase pengguna berbagai bahasa pemrograman pada tahun 2023. Kodenya pun kurang lebih sama dengan kode sebelumnya seperti Kode ini mengimpor dua modul dari pustaka *Matplotlib*, yaitu *pyplot* untuk membuat *plot* dan *style* untuk mengatur gaya *plot* dan Mengatur gaya *plot* menggunakan gaya '*ggplot*' dari *Matplotlib*. Kemudian Menyediakan data yang akan digunakan untuk membuat *plot*. Bahasa pemrograman adalah daftar nama bahasa pemrograman, dan persentase pengguna adalah persentase pengguna untuk masing-masing bahasa dan selanjutnya Membuat diagram lingkaran (*pie chart*) dengan menggunakan data yang telah disediakan. Fungsi *pie* digunakan untuk membuat *pie chart*, dengan parameter persentase pengguna, label, format persentase (*autopct*), sudut awal (*startangle*), dan warna menggunakan *plt.cm.Paired.colors* dan yang terakhir adalah menampilkan *plot* setelah konfigurasi selesai. Fungsi *tight_layout* digunakan agar elemen-elemen *plot* tidak tumpang tindih. Fungsi *show()* menampilkan *plot* secara keseluruhan.

- Luaran



Gambar 2.6 Tampilan Luaran/*Output Pie Chart*

- Penjelasan :

Gambar 2.6 diatas merupakan tampilan atau *Output* dari *source code* dan gambar 2.5 dimana didalamnya terdapat tampilan yang sebelumnya kita panggil. Yang dimana ada judul yaitu bahasa pemrograman populer (2023), 10 daftar nama yang dimasukan dan presentase penggunaanya atau berapa persen pengguna bahasa, yang juga terdapat hasil peresentase dari bahasa pemrograman yang populer pada tahun 2023 yang ditampilkan dalam bentuk lingkaran grafik tersebut.

- Gelombang Sinus

Gelombang sinusoidal adalah salah satu dari bentuk gelombang yang penting di Teknik Elektro. Gelombang sinus sangat penting dalam bidang fisika karena gelombang ini

mempertahankan bentuknya ketika ditambahkan kepada gelombang sinus berfrekuensi sama yang lain walaupun fasenya berbeda. Gelombang ini merupakan satu-satunya fungsi periodik yang memiliki sifat ini, menjadikan gelombang ini bagian penting dalam Analisis Fourier.

```

line_chart.py  gelombang_sinus.py x  pie_chart.py  grafik.py  bar_chart.py  readme.md
kode_grafik > gelombang_sinus.py > ...
1  import numpy as np
2  import matplotlib.pyplot as plt
3  import matplotlib.style as style
4
5  # Mengatur gaya plot
6  style.use('ggplot')
7
8  # Data untuk plot
9  bahasa_pemrograman = ["JavaScript", "HTML/CSS", "Python", "SQL", "TypeScript", "Bash/Shell", "Java", "C#", "C++", "PHP"]
10 persentase_pengguna = [65.82, 52.83, 51.52, 45.32, 43.75, 32.74, 30.49, 29.16, 20.21, 19.03]
11
12 # Membuat data untuk contoh gelombang sinus
13 x_sinus = np.linspace(0, 5, 100) # 100 titik antara 0 dan 5
14 y_sinus = np.sin(x_sinus) * 5 + 25 # Gelombang sinus dengan amplitudo 5 dan fase 25
15
16 # Membuat plot
17 fig, ax = plt.subplots()
18
19 # Membuat diagram garis untuk persentase pengguna
20 ax.plot(bahasa_pemrograman, persentase_pengguna, marker='o', linestyle='-', color='skyblue', label='Persentase Pengguna')
21
22 # Membuat plot gelombang sinus
23 ax.plot(x_sinus, y_sinus, linestyle='--', color='orange', label='Gelombang Sinus')
24
25 # Menambahkan label dan judul
26 ax.set_title('Bahasa Pemrograman Populer (2023) dan Gelombang Sinus')
27 ax.set_ylabel('Persentase Pengguna (%) / Nilai Gelombang Sinus')
28 ax.set_xlabel('Bahasa Pemrograman')
29
30 # Menambahkan label pada setiap titik data persentase pengguna
31 for i, v in enumerate(persentase_pengguna):
32     ax.text(i, v + 0.5, str(v), color='black', ha='center')
33
34 # Menambahkan legend
35 ax.legend()
36
37 # Menampilkan grid
38 ax.grid(True, linestyle='--', alpha=0.7)
39
40 # Menampilkan plot
41 plt.tight_layout()
42 plt.show()
43

```

Gambar 2.7 Tampilan Code Gelombang Sinus

- *Source Code*

```

import numpy as np
import Matplotlib.pyplot as plt
import Matplotlib.style as style

# Mengatur gaya plot
style.use('ggplot')

# Data untuk plot
bahasa_pemrograman = ["JavaScript", "HTML/CSS", "Python", "SQL", "TypeScript",
"Bash/Shell", "Java", "C#", "C++", "PHP"]
persentase_pengguna = [65.82, 52.83, 51.52, 45.32, 43.75, 32.74, 30.49, 29.16, 20.21, 19.03]

```

```

# Membuat data untuk contoh gelombang sinus
x_sinus = np.linspace(0, 5, 100) # 100 titik antara 0 dan 5
y_sinus = np.sin(x_sinus) * 5 + 25 # Gelombang sinus dengan amplitudo 5 dan fase 25

# Membuat plot
fig, ax = plt.subplots()

# Membuat diagram garis untuk persentase pengguna
ax.plot(bahasa_pemrograman, persentase_pengguna, marker='o', linestyle='-', color='skyblue',
label='Persentase Pengguna')

# Membuat plot gelombang sinus
ax.plot(x_sinus, y_sinus, linestyle='--', color='orange', label='Gelombang Sinus')

# Menambahkan label dan judul
ax.set_title('Bahasa Pemrograman Populer (2023) dan Gelombang Sinus')
ax.set_ylabel('Persentase Pengguna (%) / Nilai Gelombang Sinus')
ax.set_xlabel('Bahasa Pemrograman')

# Menambahkan label pada setiap titik data persentase pengguna
for i, v in enumerate(persentase_pengguna):
    ax.text(i, v + 0.5, str(v), color='black', ha='center')

# Menambahkan legend
ax.legend()

# Menampilkan grid
ax.grid(True, linestyle='--', alpha=0.7)

# Menampilkan plot
plt.tight_layout()
plt.show()

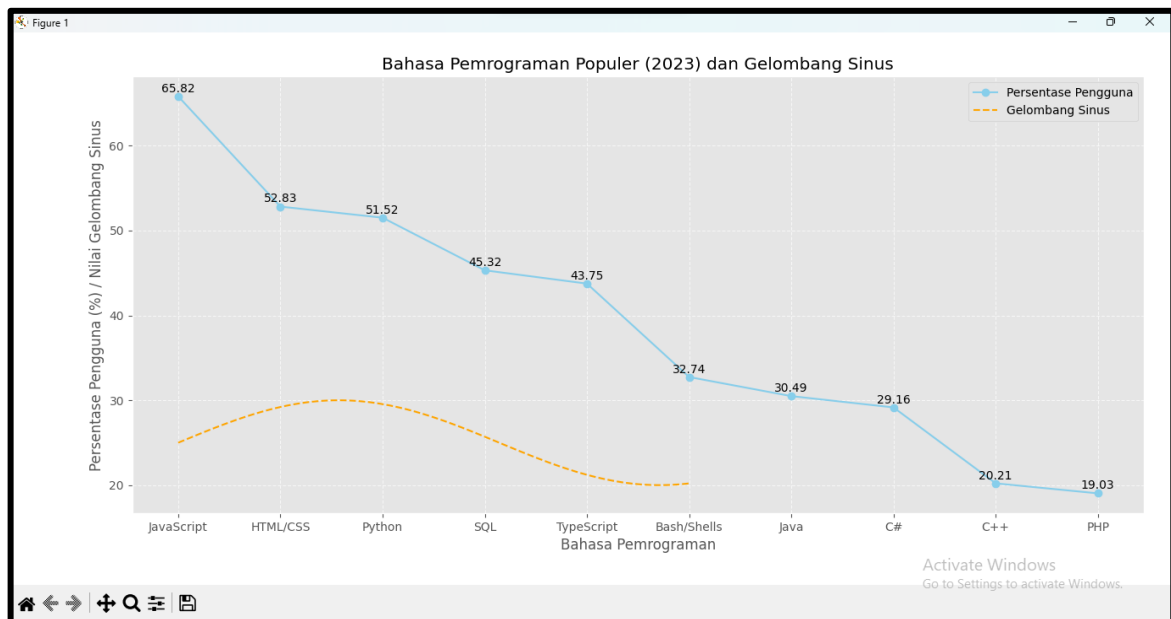
```

- Penjelasan :

Gambar 2.7 dan *Source Code* diatas adalah membuat *plot* gabungan yang menggambarkan persentase pengguna bahasa pemrograman pada tahun 2023 dan contoh

gelombang sinus. Kode ini mengimpor modul NumPy untuk manipulasi array, dan modul *Matplotlib* untuk membuat *plot* dan mengatur gaya *plot* menggunakan gaya 'ggplot' dari *Matplotlib*.. Kemudian kita dapat Menyediakan data yang akan digunakan untuk membuat *plot* persentase pengguna bahasa pemrograman dan Membuat data untuk contoh gelombang sinus dengan menggunakan NumPy dan selanjutnya Membuat objek gambar (fig) dan sumbu (ax) menggunakan fungsi *subplots()*. Kemudian kita dapat Membuat diagram garis untuk persentase pengguna bahasa pemrograman dengan menggunakan data yang telah disediakan dan Membuat *plot* gelombang sinus dengan menggunakan data yang telah dibuat sebelumnya dan juga Menambahkan judul dan label pada sumbu-x dan sumbu-y. Dan selanjutnya kita dapat menambahkan label pada setiap titik data persentase pengguna untuk menunjukkan nilai persentase dan Menambahkan legenda pada *plot* dan juga menampilkan grid pada *plot* dengan garis putus-putus. Dan selanjutnya yang terakhir menampilkan *plot* setelah konfigurasi selesai. Fungsi *tight_layout()* digunakan agar elemen-elemen *plot* tidak tumpang tindih. Fungsi *show()* menampilkan *plot* secara keseluruhan.

- Luaran



Gambar 2.8 Tampilan Luaran/Output Gelombang Sinus

- Penjelasan :

Gambar 2.8 diatas merupakan tampilan atau *Output* dari *source code* dan gambar 2.7 dimana didalamnya terdapat tampilan yang sebelumnya kita panggil. Yang dimana ada judul yaitu bahasa pemograman populer (2023), 10 daftar nama yang dimasukan dan presentase penggunaanya atau berapa persen pengguna bahasa, yang juga terdapat hasil peresentase dari bahasa pemograman yang populer pada tahun 2023 yang ditampilkan dalam bentuk gelombang sinus tersebut.

- **Grafik (Semua Grafik didalam satu kode)**


```

line_chart.py  gelombang_sinus.py  pie_chart.py  grafik.py  x  bar_chart.py  readme.md
kode_grafik > grafik.py > ...
1  import numpy as np
2  import matplotlib.pyplot as plt
3  import matplotlib.style as style
4
5  # Mengatur gaya plot
6  style.use('ggplot')
7
8  # Data untuk plot
9  bahasa_pemrograman = ["JavaScript", "HTML/CSS", "Python", "SQL", "TypeScript", "Bash/Shells", "Java", "C#", "C++", "PHP"]
10 persentase_pengguna = [65.82, 52.83, 51.52, 45.32, 43.75, 32.74, 30.49, 29.16, 20.21, 19.03]
11
12 # Membuat data untuk contoh gelombang sinus
13 x_sinus = np.linspace(0, 5, 100) # 100 titik antara 0 dan 5
14 y_sinus = np.sin(x_sinus) * 5 + 25 # Gelombang sinus dengan amplitudo 5 dan fase 25
15
16 # Membuat subplot untuk line chart
17 fig, ax_line = plt.subplots()
18
19 ax_line.plot(bahasa_pemrograman, persentase_pengguna, marker='o', linestyle='-', color='skyblue', label='Line Chart')
20 ax_line.set_title('Persentase Pengguna Bahasa Pemrograman Populer (2023)')
21 ax_line.set_ylabel('Persentase Pengguna (%)')
22 ax_line.set_xlabel('Bahasa Pemrograman')
23 ax_line.legend()
24
25 # Membuat subplot untuk pie chart
26 fig_pie, ax_pie = plt.subplots()
27
28 ax_pie.pie(persentase_pengguna, labels=bahasa_pemrograman, autopct='%1.1f%%', startangle=90, colors=plt.cm.Paired.colors)
29 ax_pie.set_title('Pie Chart')
30
31 # Membuat subplot untuk bar chart
32 fig_bar, ax_bar = plt.subplots()
33
34 ax_bar.bar(bahasa_pemrograman, persentase_pengguna, color='skyblue', label='Bar Chart')
35 ax_bar.set_title('Bar Chart')
36 ax_bar.set_ylabel('Persentase Pengguna (%)')
37 ax_bar.set_xlabel('Bahasa Pemrograman')
38 ax_bar.legend()
39
40 # Membuat subplot untuk gelombang sinus
41 fig_sinus, ax_sinus = plt.subplots()
42
43 ax_sinus.plot(x_sinus, y_sinus, linestyle='--', color='orange', label='Gelombang Sinus')
44 ax_sinus.set_title('Contoh Gelombang Sinus')
45 ax_sinus.set_ylabel('Nilai Gelombang Sinus')
46 ax_sinus.set_xlabel('Waktu')
47 ax_sinus.legend()
48
49 # Menampilkan semua plot
50 plt.tight_layout()
51 plt.show()
52

```

Gambar 2.9 Tampilan *Code* Grafik

- *Source Code*

```
import numpy as np
```

```
import Matplotlib.pyplot as plt
```

```
import Matplotlib.style as style
```

```
# Mengatur gaya plot
```

```
style.use('ggplot')
```

```
# Data untuk plot
```

```
bahasa_pemrograman = ["JavaScript", "HTML/CSS", "Python", "SQL", "TypeScript",
"Bash/Shells", "Java", "C#", "C++", "PHP"]
```

```
persentase_pengguna = [65.82, 52.83, 51.52, 45.32, 43.75, 32.74, 30.49, 29.16, 20.21, 19.03]
```

```
# Membuat data untuk contoh gelombang sinus
```

```
x_sinus = np.linspace(0, 5, 100) # 100 titik antara 0 dan 5
```

```
y_sinus = np.sin(x_sinus) * 5 + 25 # Gelombang sinus dengan amplitudo 5 dan fase 25
```

```

# Membuat subplot untuk line chart
fig, ax_line = plt.subplots()
ax_line.plot(bahasa_pemrograman, persentase_pengguna, marker='o', linestyle='-',
color='skyblue', label='Line Chart')
ax_line.set_title('Persentase Pengguna Bahasa Pemrograman Populer (2023)')
ax_line.set_ylabel('Persentase Pengguna (%)')
ax_line.set_xlabel('Bahasa Pemrograman')
ax_line.legend()

# Membuat subplot untuk pie chart
fig_pie, ax_pie = plt.subplots()
ax_pie.pie(persentase_pengguna, labels=bahasa_pemrograman, autopct='%1.1f%%',
startangle=90, colors=plt.cm.Paired.colors)
ax_pie.set_title('Pie Chart')

# Membuat subplot untuk Bar Chart
fig_bar, ax_bar = plt.subplots()

ax_bar.bar(bahasa_pemrograman, persentase_pengguna, color='skyblue', label='Bar Chart')
ax_bar.set_title('Bar Chart')
ax_bar.set_ylabel('Persentase Pengguna (%)')
ax_bar.set_xlabel('Bahasa Pemrograman')
ax_bar.legend()

# Membuat subplot untuk gelombang sinus
fig_sinus, ax_sinus = plt.subplots()
ax_sinus.plot(x_sinus, y_sinus, linestyle='--', color='orange', label='Gelombang Sinus')
ax_sinus.set_title('Contoh Gelombang Sinus')
ax_sinus.set_ylabel('Nilai Gelombang Sinus')
ax_sinus.set_xlabel('Waktu')
ax_sinus.legend()

# Menampilkan semua plot
plt.tight_layout()
plt.show()

```

- Penjelasan :

Gambar 2.9 dan *Source Code* diatas adalah membuat empat *subplot* yang menampilkan berbagai jenis *plot* berbeda, termasuk line chart, pie chart, *Bar Chart*, dan *plot* gelombang sinus. Kode ini mengimpor modul NumPy untuk manipulasi array, dan modul *Matplotlib* untuk membuat *plot* dan mengatur gaya *plot* menggunakan gaya 'ggplot' dari *Matplotlib* juga menyediakan data yang akan digunakan untuk membuat *plot* persentase pengguna bahasa pemrograman. Selanjutnya kita dapat membuat data untuk contoh gelombang sinus dengan menggunakan NumPy dan Membuat *subplot* untuk *Line chart* yang menampilkan persentase pengguna bahasa pemrograman. Dan selanjutnya kita dapat membuat *subplot* untuk *Pie Chart* yang menampilkan persentase pengguna bahasa pemrograman dan kemudian membuat *subplot* untuk *Bar Chart* yang menampilkan persentase pengguna bahasa pemrograman dan membuat *subplot* untuk *plot* gelombang sinus. Dan selanjutnya yang terakhir Menampilkan semua *subplot* secara bersamaan. Fungsi *tight_layout()* digunakan agar elemen-elemen *plot* tidak tumpang tindih. Fungsi *show()* menampilkan *plot* secara keseluruhan.

- Luaran



Gambar 2.10 Tampilan Luaran/*Output* Grafik

- Penjelasan :

Gambar 2.8 diatas merupakan tampilan atau *Output* dari source kode dan gambar 2.7 dimana didalamnya terdapat tampilan yang sebelumnya kita panggil. Yang dimana ada judul yaitu bahasa pemograman populer (2023), 10 daftar nama yang dimasukan dan presentase penggunaanya atau berapa persen pengguna bahasa, yang juga terdapat hasil peresentase dari bahasa pemograman yang populer pada tahun 2023 yang ditampilkan dalam bentuk dari semua grafik yang sebelumnya seperti line chart, pie chart, *Bar Chart*, dan *plot* gelombang sinus tersebut.

BAB III

KESIMPULAN DAN SARAN

3.1 Kesimpulan

Object oriented programming atau OOP adalah sebuah konsep programming yang berorientasi pada data atau objek untuk mengatur desain program. Program yang sudah ada merupakan gabungan dari objek-objek yang saling berhubungan dan disusun menjadi satu class. Dengan begitu, OOP akan membantu *Developer* menyelesaikan sejumlah masalah program yang cukup rumit. Definisi lain dari OOP yaitu konsep yang mengidentifikasi object class terkait metode yang digunakan dalam pemrograman. Konsep OOP menawarkan potensi development menuju tingkat abstraksi lebih tinggi.

Python adalah bahasa pemrograman yang banyak digunakan dalam aplikasi web, pengembangan perangkat lunak, ilmu data, dan machine learning (ML). *Developer* menggunakan *Python* karena efisien dan mudah dipelajari serta dapat dijalankan di berbagai platform. Perangkat lunak *Python* dapat diunduh secara gratis, terintegrasi baik dengan semua tipe sistem, dan meningkatkan kecepatan pengembangan.

Dengan menggunakan *Python* kita dapat membuat grafik apapun dengan mudah, dan untuk membuatnya *Library*/pustaka *Python*, dalam membuat penggunaan grafik pada *Python*, dalam membuat grafik ada beberapa ada tahapan *Library* yang dapat digunakan misalnya *Matplotlib*, *Visualisasi* *Panda*, *Seaborn*, *ggplot* dan *Plotly* dll. Disini kami menggunakan *Matplotlib* untuk penggunaannya. Dan menambahkan kode kode perintah seperti diatas pada pembahasan.

3.2 Saran

Pada pembuatan tugas ini kami menyarankan sebelum menggunakan kode pada *pyhton* dalam oop sangat harus memahaminya terlebih dahulu karena jika tidak paham ap aitu *pyhton* dalam oop maka pada saat proses pembuatan akan terjadi error atau tidak bisa di jalankan.

DAFTAR PUSTAKA

Aws.Amazon (2022) :*"Apa Itu Python?"*

<https://aws.amazon.com/id/what-is/Python/>

(Diakses pada 12 desember 2023)

Dewa web, September 15, 2022: *"Mengenal Apa itu OOP, Prinsip, Kelebihan, dan Kekurangannya"*

<https://www.dewaweb.com/blog/apa-itu-oop/>

(Diakses pada 12 desember 2023)

Skodev (2023) *"10 Bahasa Pemrograman Terpopuler (Lengkap Survey 2023)"*

<https://sko.dev/bahasa-pemrograman-terpopuler>

(Diakses pada 12 desember 2023)



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BENGKULU
FAKULTAS TEKNIK
PROGRAM STUDI SISTEM INFORMASI

Jalan W.R. Supratman Kandang Limun Bengkulu

Bengkulu 38371 A Telepon : (0736) 344087, 22105 – 227

LEMBAR ACC PROYEK PEMEROGRAMAN BERORIENTASI OBJEK

Nama Mahasiswa : 1. Novia Al Qodri (G1F022015)
2. Suryaningsih (G1F022049)
3. F Yoshua Habeahan (G1F022025)

Dosen : 1. Ferzha Putra Utama, S.T.,M.Eng
2. Arie Vatesia, S.T.,M.TI,Ph.D.

Asisten : 1. Rendi Julian Saputra (G1A019066)

Laporan Praktikum Hari/Tanggal Laporan diberikan	Tanda ACC Hari/Tanggal Laporan di ACC
Laporan Tugas Akhir	