

LAPORAN TUGAS UAS
PEMROGRAMAN BERORIENTASI OBJEK



DISUSUN OLEH :

Yuma Juliana (G1F022003)

Chindy Feby Amara (G1F022045)

Nama Asisten Dosen :

Randi Julian Saputra

Dosen Pengampu :

Arie Vatesia, S.T.,M.TI,Ph.D

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS BENGKULU
2023/2022

KATA PENGANTAR

Puji Syukur atas kehadiran Allah SWT yang memberikan Rahmat dan hidayah-Nya sehingga kami dapat menyelesaikan laporan tugas UAS Pemrograman Berorientasi Objek. Tugas ini disusun dengan tujuan untuk melengkapi tugas uas dan penerapan konsep-konsep dasar dalam paradigma pemrograman berorientasi objek yang telah kami pelajari selama semester ini. Pemrograman berorientasi objek merupakan pendekatan yang memberikan kemudahan dalam mengorganisasi dan mengelola kode program dengan memandangnya sebagai suatu kumpulan objek yang memiliki sifat, perilaku, dan hubungan di antara satu sama lain. Dalam tugas ini, kami mencoba menggali lebih dalam konsep-konsep tersebut melalui pembuatan program-program sederhana yang merefleksikan pemahaman kami terhadap konsep dasar seperti pewarisan, enkapsulasi, dan polimorfisme.

Bengkulu, 13 Desember 2023

Penyusun

DAFTAR ISI

KATA PENGANTAR	1
DAFTAR ISI	2
BAB I PENDAHULUAN	3
1.1 Latar Belakang	3
1.2 Rumusan Masalah.....	4
1.3 Tujuan	4
1.4 Manfaat	4
BAB II LANDASAN TEORI.....	5
2.1 Sejarah Bahasa Pemrograman	5
2.2 Pemrograman berorientasi objek.....	6
2.3 Karakteristik Pemrograman Berorientasi Objek	6
2.3.1 Keuntungan Pemrograman Berorientasi Objek.....	7
2.4 Phyton	8
2.4.1 Sejarah Phyton	8
2.4.2 Pengertian Phyton.....	9
2.4.3 Penggunaan Phyton	9
2.5 Karakteristik Pemrograman Berorientasi Objek	10
BAB III SOAL DAN PEMBAHASAN	11
3.1 SOAL	11
3.2 PEMBAHASAN.....	11
BAB IV KESIMPULAN DAN SARAN	27
4.1 KESIMPULAN.....	27
4.2 SARAN	27
DAFTAR PUSTAKA.....	28

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pemrograman berorientasi objek membantu Kita untuk memahami dan merepresentasikan dunia nyata dengan cara yang lebih baik. Konsep seperti pewarisan, enkapsulasi, dan polimorfisme memungkinkan Anda memodelkan objek dan hubungannya dengan objek lain secara lebih alami.

Pemrograman berorientasi objek (object-oriented programming disingkat OOP) merupakan paradigma pemrograman berdasarkan konsep "objek", yang dapat berisi data, dalam bentuk field atau dikenal juga sebagai atribut; serta kode, dalam bentuk fungsi/prosedur atau dikenal juga sebagai method. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya, de dasar pada OOP adalah mengkombinasikan data dan fungsi untuk mengakses data menjadi sebuah kesatuan unit yang dikenal dengan nama objek. Objek adalah struktur data yang terdiri dari bidang data dan metode bersama dengan interaksi mereka untuk merancang aplikasi dan program komputer. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objeklainnya.

Pemrograman berorientasi objek dalam melakukan pemecahan suatu masalah tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Sebagai contoh sebuah departemen yang memiliki seorang manager, sekretaris, petugas administrasi data dan lainnya. Jika manager ingin memperoleh data dari bagian administrasi maka manager tersebut tidak harus mengambilnya langsung tetapi dapat menyuruh petugas bagian administrasi untuk mengambilnya. Pada kasus tersebut seorang manager tidak harus mengetahui bagaimana cara mengambil data tersebut tetapi manager bisa mendapatkan data tersebut melalui objek petugas administrasi. Jadi untuk menyelesaikan suatu masalah dengan kolaborasi antar objek-objek yang ada karena setiap objek memiliki deskripsi tugasnya sendiri.

Pemrograman berorientasi objek bekerja dengan baik ketika dibarengi dengan Objek-Oriented Analysis And Design Process (OOAD). Jika membuat program berorientasi objek tanpa OOAD, seperti membangun rumah tanpa terlebih dahulu penganalisis apa saja yang dibutuhkan oleh rumah itu, tanpa perencanaan, tanpa blue-print, tanpa menganalisis ruangan apa saja yang diperlukan, beberapa besar rumah yang akan dibangun dan sebagainya.

1.2 Rumusan Masalah

Permasalahan yang akan di bahas di dalam laporan tugas UAS proyek pemrograman berorientasi objek ini adalah sebagai berikut.

- 1.2.1 Membuat program untuk mengedit gambar dengan phyton.
- 1.2.2 Membuat program untuk memotong gambar dengan phyton.
- 1.2.3 Membuat program untuk membuka gambar dengan phyton.
- 1.2.4 Membuat program untuk mengubah ukuran gambar dengan phyton.
- 1.2.5 Membuat program untuk menyimpan gambar dengan phyton.

1.3 Tujuan

Tujuan yang akan di Bahas di dalam laporan tugas uas pemrograman berorientasi objek ini adalah sebagai berikut.

- 1.3.1 Untuk membuat program untuk mengedit gambar dengan phyton.
- 1.3.2 Untuk membuat program untuk memotong gambar dengan phyton.
- 1.3.3 Untuk membuat program untuk membuka gambar dengan phyton.
- 1.3.4 Untuk membuat program untuk mengubah ukuran gambar dengan phyton.
- 1.3.5 Untuk membuat program untuk menyimpan gambar dengan phyton.

1.4 Manfaat

1.4.1 Manfaat Bagi Mahasiswa

Manfaat yang dapat di rasakan atau digunakan oleh mahasiswa dalam melakukan pembuatan laporan tugas uas ini adalah mahasiswa dapat meningkatkan wawasan atau pengetahuan serta pengalaman tentang pembuatan program mengedit gambar menggunakan phyton.

1.4.2 Manfaat Bagi Pembaca

Manfaat yang dapat dirasakan oleh para pembaca adalah pembaca dapat menjadikan laporan tugas uas ini sebagai referensi dalam mempelajari pemrograman berorientasi objek. Mulai dari pembuatan kode untuk mengedit gambar menggunakan phyton mereka dapat memanfaatkan laporan ini untuk melengkapi tugas dan lain lain.

1.4.3 Manfaat Bagi Dosen dan Asisten Dosen

Manfaat yang dapat di rasakan oleh dosen dan asisten dosen adalah dosen dan asisten dosen dapat mengetahui sejauh mana pengetahuan mahasiswa dalam pembuatan program untuk mengedit gambar menggunakan phyton dan menjadi acuan untuk memberikan nilai terhadap mahasiswa.

BAB II

LANDASAN TEORI

2.1 Sejarah Bahasa Pemrograman

Komputer tidak dapat mengerti dengan mudah instruksi yang dimasukkan oleh manusia. Komputer butuh alat untuk menerjemahkan bahasa manusia kedalam bahasa “mesin”. Istilah “mesin” digunakan karena pada dasarnya komputer adalah mesin yang dapat menerima instruksi, memproses, dan mengeluarkan keluaran output kedalam perangkat lain yang terhubung (monitor). Alat yang digunakan untuk memudahkan komputer memahami instruksi yang dimasukkan adalah bahasa pemrograman. Ada banyak sekali bahasa pemrograman yang dikembangkan untuk mempermudah manusia dalam memprogram di komputer. Komputer memiliki bahasa dasar (native) yang dikenal dengan bahasa mesin. Bahasa mesin terdiri dari rangkaian urutan angka biner. Berikut adalah contoh penulisan bahasa mesin: 1011101010111010101011110111110111101 Dapat dibayangkan ketika programmer harus memprogram dengan bahasa mesin, hal tersebut sangatlah tidak menyenangkan. Program yang ditulis akan sulit untuk dibaca dan dimodifikasi.

Dengan alasan tersebut muncul bahasa pemrograman assembly. Assembly memungkinkan programmer untuk memasukkan instruksi dengan kata-kata singkat atau tidak lagi rangkaian angka biner. Kata-kata singkat tadi juga dikenal dengan istilah “mnemonic”. Berikut adalah contoh kode yang berisi instruksi sederhana dengan bahasa assembly: add 2, 3, result Menulis dengan bahasa assembly lebih mudah dibandingkan dengan bahasa mesin. Meskipun demikian, bahasa ini merupakan bahasa yang paling dekat dengan bahasa mesin. Untuk memprogram dengan assembly programmer harus sudah memahami cara kerja CPU secara terperinci.

Pada tahun 1950-an, generasi baru pemrograman muncul dengan kategori bahasa tingkattinggi. Bahasa tingkat-tinggi mendukung penulisan program dengan bahasa inggris dan mudah untuk digunakan. Selain itu, bahasa tingkat-tinggi ini dapat dijalankan dibeberapa jenis mesin (tidak hanya pada satu mesin tertentu). Instruksi-instruksi perintah yang 7 dimasukkan dengan bahasa tingkat-tinggi ini dikenal dengan istilah statement. Berikut adalah contoh statement dengan bahasa tingkat-tinggi untuk menghitung luas lingkaran: $area = 7 * 7 * 3.14$; Python Bahasa pemrograman yang dapat berguna untuk apa saja (genera purpose language). Sangat efektif untuk menulis program-program singkat Awal Pengembangan (1989-1991): Python pertama kali dikembangkan oleh Guido van Rossum pada akhir 1980-an di Centrum Wiskunde & Informatica (CWI) di Belanda. Versi pertama Python, Python 0.9.0, dirilis pada Februari 1991. Nama "Python" diambil dari acara komedi Inggris "Monty Python's Flying Circus," yang merupakan favorit van Rossum.

2.2 Pemrograman berorientasi objek

Pemrograman berorientasi objek (*object-oriented programming* disingkat OOP) merupakan paradigma pemrograman berdasarkan konsep "objek", yang dapat berisi data, dalam bentuk *field* atau dikenal juga sebagai atribut; serta kode, dalam bentuk fungsi/prosedur atau dikenal juga sebagai *method*. Semua data dan fungsi di dalam paradigma ini dibungkus dalam *kelas-kelas* atau *objek-objek*. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya, Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik peranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

2.3 Karakteristik Pemrograman Berorientasi Objek

1. Abstrak

Abstrak artinya Mengabaikan hal-hal dalam topik yang tidak terkait dengan tujuan saat ini dan fokus pada aspek yang terkait dengan tujuan saat ini. Ini untuk mengekstrak jenis hal tertentu di dunia nyata dan mengekspresikannya dengan kode program. Abstrak umumnya disebut kelas atau antarmuka. Abstraksi tidak bermaksud untuk memahami semua masalah, tetapi untuk memilih beberapa di antaranya, tanpa beberapa detail untuk saat ini. Abstraksi mencakup dua aspek, abstraksi data, tetapi abstraksi proses. Abstraksi data mewakili karakteristik kelas benda di dunia, yaitu atribut objek. Misalnya, burung memiliki sayap, bulu, dll. Atau juga disebut atribut kelas.

Abstraksi proses mewakili perilaku kelas benda di dunia, yaitu, perilaku objek. Misalnya, burung dapat terbang dan berkicau. Atau disebut dengan metode kelas

2. Paket

Enkapsulasi merupakan salah satu ciri dari berorientasi objek, dan ciri utama dari konsep objek dan kelas. Enkapsulasi adalah untuk mengelilingi proses dan data, dan akses ke data hanya dapat melalui antarmuka yang ditentukan. Seperti variabel pribadi, gunakan set dan dapatkan metode untuk mendapatkannya. Enkapsulasi memastikan bahwa modul memiliki independensi yang baik, membuat pemeliharaan dan modifikasi program lebih mudah. Modifikasi program aplikasi terbatas pada bagian dalam kelas. Sehingga dampak modifikasi program aplikasi dapat diminimalkan.

3. Warisan

Model hierarkis untuk menghubungkan kelas, memungkinkan dan mendorong penggunaan kembali kelas, dan menyediakan cara untuk mengekspresikan kesamaan dengan jelas. Kelas baru dari objek dapat diturunkan dari kelas yang sudah ada. Proses ini disebut pewarisan kelas. Kelas baru mewarisi karakteristik kelas asli, kelas baru disebut kelas turunan (subclass) dari kelas asli, dan kelas asli disebut kelas dasar (kelas induk) dari kelas baru. Kelas turunan dapat mewarisi metode dan variabel instan dari kelas induknya dan kelas dapat memodifikasi atau menambahkan metode baru agar lebih cocok untuk kebutuhan khusus. Oleh karena itu, dapat dikatakan bahwa pewarisan adalah menggunakan kembali kode kelas induk dan sekaligus mempersiapkan realisasi polimorfisme. Baca juga Dasar Pemrograman Berorientasi Objek Di Pemrograman C++

4. Polimorfisme

Polimorfisme mengacu pada memungkinkan berbagai jenis objek untuk menanggapi pesan yang sama. Polimorfisme meliputi polimorfisme parametrik dan polimorfisme inklusi. Bahasa polimorfik memiliki keunggulan fleksibilitas, abstraksi, berbagi perilaku, berbagi kode, yang memecahkan masalah nama fungsi aplikasi yang sama. Secara umum, penulisan ulang metode, kelebihan beban, dan tautan dinamis merupakan polimorfisme. Salah satu alasan mengapa java memperkenalkan konsep polimorfisme adalah untuk menutupi kekurangan fungsionalitas yang disebabkan oleh pewarisan kelas tunggal.

2.3.1 Keuntungan Pemrograman Berorientasi Objek

Ada beberapa keuntungan yang bisa didapat berkat pemrograman berorientasi objek. Di antara mereka perlu disebutkan:

1. Dapat dimodifikasi yang memungkinkan untuk menambah, mengubah, atau menghapus objek atau fungsi baru dan ini memudahkan pembaruan program.
2. Manajemen kesalahan, ketika pemrograman berorientasi objek digunakan lebih mudah untuk menemukan momen ketika kesalahan terjadi.
3. Lebih mudah untuk melakukan kerja kelompok berkat enkapsulasi yang dimiliki oleh jenis pemrograman ini.
4. Kita dapat membuat kelas dan kemudian membagikan propertinya dengan subkelas lain, dengan kata lain, kelas tersebut memiliki warisan. Dengan program ini, biaya pemrograman dapat dikurangi secara signifikan.
5. Kerangka kerja yang sesuai dapat dibuat yang membuatnya lebih mudah untuk membuat antarmuka grafis. Dapat dipasangkan dengan sangat baik dengan database.

2.3.2 Kekurangan Pemrograman Berorientasi Objek

1. Beberapa kelemahannya disebutkan di bawah ini:
2. Program berorientasi objek mungkin sedikit lebih lambat.
3. Mereka tidak memiliki kapasitas untuk diubah sepenuhnya.
4. Semua objek yang berpartisipasi membutuhkan dokumentasi dalam jumlah besar dan, karena objeknya abstrak, mungkin timbul masalah karena objek tersebut tidak sesuai dengan visi dua pemrogram yang berbeda.
5. Ketika bahasa yang digunakan dalam pemrograman berorientasi objek dipaksakan, hilangnya beberapa karakteristik yang dimiliki oleh bahasa yang bermanfaat dapat terjadi.
6. Ada kemungkinan programmer tidak mengetahui beberapa karakteristik yang dimiliki paradigma.
7. Juga tidak ada cara yang baku untuk menyelesaikan masalah, yang dapat menimbulkan berbagai jenis penafsiran terkait dengan solusi yang diajukan.

2.4 Phyton

Sejarah Phyton

Python dibuat pada akhir 1980-an[34] oleh Guido van Rossum di Centrum Wiskunde & Informatica (CWI) di Belanda sebagai penerus bahasa ABC (sendiri terinspirasi oleh SETL),[35] mampu menangani pengecualian dan berinteraksi dengan sistem operasi Amoeba.[36] Implementasinya dimulai pada bulan Desember 1989. Van Rossum memikul tanggung jawab penuh atas proyek tersebut, sebagai pengembang utama, hingga 12 Juli 2018, ketika ia mengumumkan "liburan permanen" dari tanggung jawabnya sebagai Benevolent Dictator For Life Python, sebuah gelar yang diberikan komunitas Python kepadanya untuk mencerminkan komitmen jangka panjangnya sebagai pengambil keputusan utama proyek.

Python 2.0 dirilis pada 16 Oktober 2000 dengan banyak fitur utama baru, termasuk pengumpul sampah pendeteksi siklus dan dukungan untuk Unicode.

Tanggal akhir masa pakai Python 2.7 yang awalnya ditetapkan pada tahun 2015 kemudian ditunda hingga tahun 2020 karena sejumlah besar kode yang tidak dapat dengan mudah dilanjutkan ke Python 3. Python 3.6 (dan setiap perilisan lama), tidak lagi didukung per 2021.

Pada 2022, Python 3.10.4 dan 3.9.12 dipercepat dan begitu juga perilisan yang lebih lama termasuk 3.8.13, dan 3.7.13 dikarenakan banyak masalah keamanan pada 2022. Python 3.9.13

adalah versi 3.9 terbaru, dan mulai sekarang 3.9 (dan yang terlama; 3.8 dan 3.7) hanya akan mendapatkan pembaruan keamanan.

Pengertian Phyton

Python adalah bahasa pemrograman tujuan umum yang ditafsirkan, tingkat tinggi. Dibuat oleh Guido van Rossum dan pertama kali dirilis pada tahun 1991, filosofi desain Python menekankan keterbacaan kode dengan penggunaan spasi putih yang signifikan. Konstruksi bahasanya dan pendekatan berorientasi objek bertujuan untuk membantu pemrogram menulis kode yang jelas dan logis untuk proyek skala kecil dan besar. Python diketik secara dinamis dan pengumpulan sampah. Ini mendukung beberapa paradigma pemrograman, termasuk pemrograman terstruktur (terutama, prosedural), berorientasi objek, dan fungsional. Python sering dideskripsikan sebagai bahasa "termasuk baterai" karena perpustakaan standarnya yang komprehensif.

Python dibuat pada akhir 1980-an sebagai penerus bahasa ABC. Python 2.0, dirilis pada tahun 2000, memperkenalkan fitur-fitur seperti pemahaman daftar dan sistem pengumpulan sampah dengan penghitungan referensi. Python 3.0, dirilis pada tahun 2008, adalah revisi utama dari bahasa yang tidak sepenuhnya kompatibel dengan versi sebelumnya, dan banyak kode Python 2 yang tidak berjalan tanpa modifikasi pada Python 3. Penerjemah Python tersedia untuk banyak sistem operasi. Komunitas pemrogram global mengembangkan dan memelihara CPython, implementasi referensi yang bebas dan sumber terbuka. Sebuah organisasi nirlaba, Python Software Foundation, mengelola dan mengarahkan sumber daya untuk pengembangan Python dan CPython. Python secara konsisten menempati peringkat sebagai salah satu bahasa pemrograman paling populer.

Penggunaan Phyton

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.

Untuk memahami lebih detail contoh penggunaannya, simak penjelasan berikut ini!

1. Data Analisis dan Machine Learning

Python telah menjadi andalan dalam ilmu data. Bahasa pemrograman ini memungkinkan analisis data untuk melakukan perhitungan statistik yang rumit, membuat visualisasi data serta algoritma machine learning. Ia juga bisa digunakan untuk memanipulasi, menganalisis data, dan menyelesaikan berbagai tugas lain terkait data. Selain itu, ia bisa membantu membangun berbagai visualisasi data yang berbeda. Misalnya, grafik garis dan batang, diagram lingkaran, histogram, dan lain sebagainya.

2. Pengembangan Web (Web Development)

Python juga sering digunakan untuk mengembangkan back-end sebuah situs atau aplikasi (bagian yang tidak dilihat oleh pengguna). Perannya dalam pengembangan web termasuk pengiriman data dari dan ke server, memproses data dan berkomunikasi dengan server, perutean URL serta memastikan keamanan situs. Selain itu, ia juga menawarkan beberapa framework untuk pengembangan web seperti Flask dan Django.

3. Otomatisasi Tugas

Jika kamu bosan melakukan tugas yang berulang, kamu bisa mengefisienkan waktu dengan Python. Penulisan kode yang digunakan untuk membangun proses otomatis ini disebut dengan scripting. Dalam dunia coding, otomatisasi ini bisa digunakan untuk memeriksa kesalahan di banyak file sekaligus, mengonversi file, menjalankan matematika sederhana serta menghapus duplikat dalam data.

Bahasa pemrograman ini bahkan bisa digunakan oleh pemula untuk mengotomatiskan tugas-tugas sederhana di komputer. Misalnya, mengganti nama file, mencari atau mengunduh konten online hingga mengirimkan email atau pesan dengan interval waktu yang diinginkan.

4. Menyelesaikan Berbagai Tugas Harian

Python tidak hanya milik para programmer dan data scientist saja. Mempelajarinya bisa memberikan peluang baru bagi profesi lain. Beberapa tugas yang bisa kamu otomatisasi dengan bahasa pemrograman ini antara lain:

- Memantau pasar saham atau harga kripto.
- Mengirimkan dirimu sendiri pesan pengingat untuk membawa payung setiap kali hujan.
- Memperbarui daftar belanjaan bahan makanan di rumah.
- Mengganti nama file dalam jumlah banyak.
- Mengubah file teks menjadi spreadsheet dan masih banyak lagi.

2.5 Karakteristik Pemrograman Berorientasi Objek

Di antara karakteristik utama yang dibedakan dalam pemrograman objek adalah sebagai berikut:

- Memiliki abstraksi, dengan kata lain, dapat menunjukkan karakteristik utama suatu objek dan menangkap perilakunya.
- Anda dapat mengumpulkan elemen-elemen yang Anda anggap relevan sehubungan dengan entitas yang sama dan yang memiliki tingkat abstraksi yang sama, karakteristik ini dikenal sebagai enkapsulasi.
- Ini bekerja di bawah prinsip penyembunyian, yang mengatakan bahwa setiap objek diisolasi dari luar dalam modul alami.
- Ini memiliki polimorfisme, ini berarti, berbagai jenis perilaku dalam objek.
- Objek dapat mewarisi properti dan perilaku dari kelas tempat mereka berada.

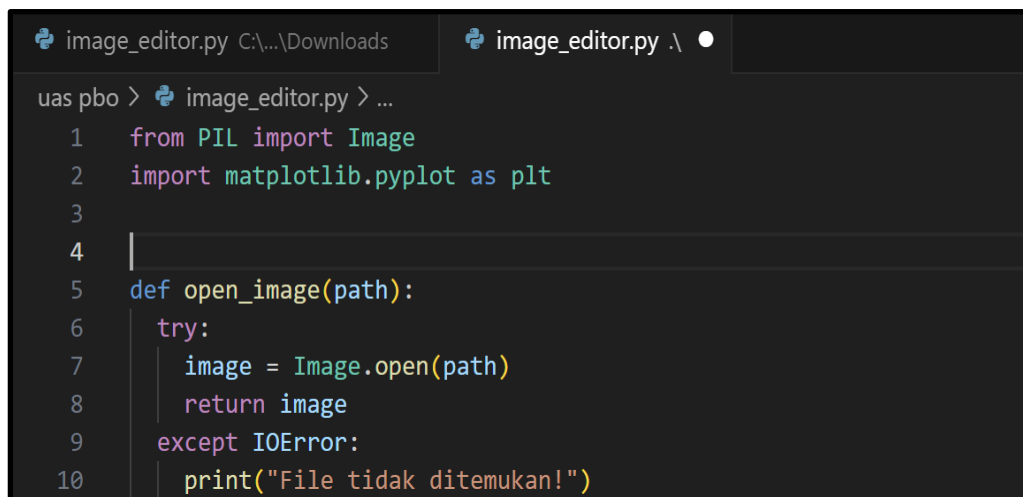
BAB III SOAL DAN PEMBAHASAN

3.1 SOAL

1. Mengedit gambar dengan Python: Perpustakaan Pencitraan Python menawarkan juru bahasa untuk mengedit gambar. Ini membantu untuk membuka, memotong, mengubah ukuran, mengambil ukuran dan menyimpan gambar.

3.2 PEMBAHASAN

A. Kode untuk membuka dan menampilkan gambar

A screenshot of a code editor with a dark background. The editor shows a file named 'image_editor.py' located in 'C:\...\Downloads'. The code is as follows:

```
1 from PIL import Image
2 import matplotlib.pyplot as plt
3
4
5 def open_image(path):
6     try:
7         image = Image.open(path)
8         return image
9     except IOError:
10        print("File tidak ditemukan!")
```

Gambar 3.1 Kode untuk membuka dan menampilkan gambar

Source code :

```
from PIL import Image
import matplotlib.pyplot as plt
# Fungsi untuk membuka dan menampilkan gambar
def open_image(path):
    try:
        image = Image.open(path)
        return image
    except IOError:
        print("File tidak ditemukan!")
```

Penjelasan kode :

from PIL import Image import matplotlib.pyplot as plt Kode ini di gunakan untuk mengimpor dua pustaka: PIL (Pillow) untuk memanipulasi gambar dan Matplotlib untuk menampilkan gambar. Yaitu dengan cara menginstal pillow pada terminal yaitu dengan cara mengetik pip install pillow maka secara otomatis akan terinstal kemudian menginstal matplotlib di terminal dengan cara mengetik pip install matplotlib maka akan terinstall.

Fungsi `def open_image(path):` `try:` menerima path file sebagai argumen. Dalam blok `try`, ini akan mencoba membuka gambar pada path yang diberikan menggunakan `Image.open` dari PIL. Jika berhasil, gambar dibaca dan dikembalikan. Jika terjadi `IOError` (jika file tidak ditemukan), pesan "File tidak ditemukan!" dicetak di konsol.

B.Kode untuk menampilkan gambar :

```
def show_image(image):  
    if image:  
        plt.imshow(image)  
        plt.show()
```

Gambar 3.2 kode menampilkan gambar

Source code :

```
def show_image(image):  
    if image:  
        plt.imshow(image)  
        plt.show()
```

Penjelasan kode :

`def show_image(image):` Fungsi ini menerima satu parameter, yaitu `image`. Parameter ini seharusnya berisi objek gambar yang dapat ditangani oleh Matplotlib. `if image:` Pada bagian ini, dilakukan pengecekan apakah parameter `image` memiliki nilai yang tidak nol atau tidak kosong. Jika `image` memiliki nilai, maka blok berikutnya akan dieksekusi; jika tidak, fungsi akan keluar tanpa melakukan apa pun. `plt.imshow(image)` `plt.show()` kode ini artinya jika `image` ada (nilai tidak nol), fungsi ini menggunakan Matplotlib untuk menampilkan gambar. `plt.imshow(image)` digunakan untuk menampilkan gambar, dan `plt.show()` digunakan untuk menampilkan plot grafik (gambar) ke layar.

C Kode untuk memotong gambar

```
def crop_image(image, top, left, bottom, right):  
    try:  
        cropped_image = image.crop((left, top, right, bottom))  
        return cropped_image  
    except ValueError:  
        print("Batas koordinat tidak valid!")
```

Gambar 3.3 kode memotong gambar

Source code :

```
def crop_image(image, top, left, bottom, right):
```

```
try:
```

```
    cropped_image = image.crop((left, top, right, bottom))
```

```
    return cropped_image
```

```
except ValueError:
```

```
    print("Batas koordinat tidak valid!")
```

Penjelasan kode :

`def crop_image(image, top, left, bottom, right):` Fungsi ini mengambil lima parameter: `image`: Objek gambar yang akan dipotong (harus dapat ditangani oleh metode `crop` dari PIL). `top`, `left`, `bottom`, `right`: ini di gunakan untuk menentukan batas koordinat untuk pemotongan gambar. Parameter ini menentukan area yang akan dipotong dari gambar yaitu dengan memasukkan memasukkan bastas atas, batas kiri, batas bawah dan batas kanan sesuai dengan ukuran yang kita inginkan maka gambarnya akan terpotong sesuai dengan instruksi yang telah kita masukkan. `cropped_image = image.crop((left, top, right, bottom))` `return cropped_image` Fungsi kode ini adalah mencoba untuk melakukan pemotongan gambar menggunakan metode `crop` dari objek gambar (`image`). Batas koordinat dipass sebagai tupel (`left`, `top`, `right`, `bottom`) ke metode `crop`. Jika pemotongan berhasil, gambar hasil pemotongan (`cropped_image`) dikembalikan.

`except ValueError: print("Batas koordinat tidak valid!")` ika terjadi `ValueError`, yang mungkin terjadi jika batas koordinat tidak valid (misalnya, `top` lebih besar dari `bottom` atau `left` lebih besar dari `right`), maka pesan kesalahan "Batas koordinat tidak valid!" dicetak di konsol.

D.Kode untuk mengubah ukuran gambar

```
def resize_image(image, width, height):  
    try:  
        resized_image = image.resize((width, height))  
        return resized_image  
    except ValueError:  
        print("Ukuran tidak valid!")
```

Gambar 3.4 kode mengubah ukuran gambar

Source code :

```
def resize_image(image, width, height):  
    try:  
        resized_image = image.resize((width, height))
```

```

return resized_image
except ValueError:
    print("Ukuran tidak valid!")

```

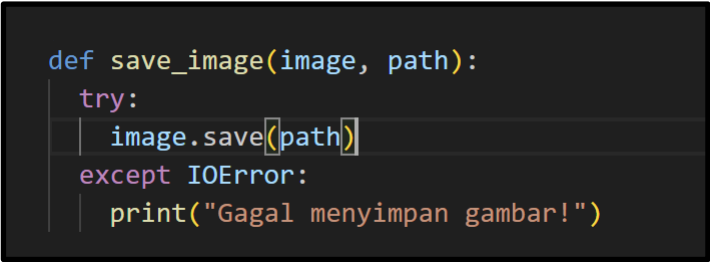
Penjelasan kode :

def resize_image(image, width, height): Fungsi kode ini adalah untuk mengambil tiga parameter: **image**: Objek gambar yang akan diubah ukurannya (harus dapat ditangani oleh metode **resize** dari PIL). **Width** Lebar baru yang diinginkan untuk gambar. **height** Tinggi baru yang diinginkan untuk gambar.

try: resized_image = image.resize((width, height)) return resized_image Fungsi kode ini adalah untuk mengubah ukuran gambar menggunakan metode **resize** dari objek gambar (**image**). Parameter (**width, height**) adalah ukuran yang baru diinginkan. Jika perubahan ukuran berhasil, gambar hasil diubah ukuran (**resized_image**) dikembalikan.

except ValueError: print("Ukuran tidak valid!") kode ini digunakan untuk Jika terjadi **ValueError**, yang mungkin terjadi jika ukuran yang dimasukkan tidak valid (misalnya, nilai negatif), maka pesan kesalahan "Ukuran tidak valid!" dicetak di konsol.

E. Kode untuk menyimpan gambar



```

def save_image(image, path):
    try:
        image.save(path)
    except IOError:
        print("Gagal menyimpan gambar!")

```

Gambar 3.5 Kode menyimpan gambar

Source code :

```

def save_image(image, path):
    try:
        image.save(path)
    except IOError:
        print("Gagal menyimpan gambar!")

```

Penjelasan :

def save_image(image, path): Kode ini di gunakan untuk fungsi **save_image** yang mengambil dua parameter, yaitu **image** (gambar yang akan disimpan) dan **path** (lokasi tempat gambar akan disimpan). **try:** Ini adalah blok **try**, yang digunakan untuk mengeksekusi potongan kode di dalamnya. Dalam konteks ini, kode di dalam **try** adalah **image.save(path)**.

`image.save(path)` Ini adalah metode dari objek gambar yang digunakan untuk menyimpan gambar ke lokasi yang ditentukan oleh `path`. Fungsi ini digunakan untuk menyimpan gambar ke lokasi yang diberikan. **except IOError:** Jika terjadi kesalahan saat menyimpan gambar, blok `except` akan menangkap kesalahan dengan tipe `IOError`. `IOError` umumnya terjadi ketika ada masalah dengan operasi input/output, seperti gagal membuka atau menulis ke file. `print("Gagal menyimpan gambar!")` kode ini di gunakan untuk jika terjadi kesalahan (`IOError`), maka pesan "Gagal menyimpan gambar!" akan dicetak ke konsol. Ini adalah cara memberikan informasi kepada pengguna bahwa ada masalah dalam menyimpan gambar.

F. Daftar menu

```
|  
print("Selamat datang di program pengeditan gambar!")  
print("1. Buka gambar")  
print("2. Potong gambar")  
print("3. Ubah ukuran gambar")  
print("4. Simpan gambar")  
print("5. Keluar")
```

Gambar 3.6 Daftar menu

Source code:

```
print("1. Buka gambar")  
print("2. Potong gambar")  
print("3. Ubah ukuran gambar")  
print("4. Simpan gambar")  
print("5. Keluar")
```

Penjelasan kode :

`print("1. Buka gambar")` Kode ini di gunakan untuk mencetak teks "1. Buka gambar" ke konsol. Ini adalah pilihan menu pertama yang menunjukkan opsi untuk membuka sebuah gambar. `print("2. Potong gambar")` Kode ini di gunakan untuk mencetak teks "2. Potong gambar" ke konsol. Ini adalah pilihan menu kedua yang menunjukkan opsi untuk melakukan operasi pemotongan pada gambar. `print("3. Ubah ukuran gambar")` Kode Ini di gunakan untuk mencetak teks "3. Ubah ukuran gambar" ke konsol. Ini adalah pilihan menu ketiga yang menunjukkan opsi untuk mengubah ukuran gambar. `print("4. Simpan gambar")` Kode ini di gunakan untuk mencetak teks "4. Simpan gambar" ke konsol. Ini adalah pilihan menu keempat yang menunjukkan opsi untuk menyimpan gambar. `print("5. Keluar")` Kode ini di gunakan untuk mencetak teks "5. Keluar" ke konsol. Ini adalah pilihan menu kelima yang menunjukkan opsi untuk keluar dari program atau aplikasi.

Jadi, secara keseluruhan, kode ini adalah bagian dari antarmuka pengguna teks yang memberikan beberapa opsi kepada pengguna, seperti membuka gambar, melakukan operasi pemotongan, mengubah ukuran gambar, menyimpan gambar, dan keluar dari program.

G.Membuka gambar

```
image = None
# Di dalam loop, setelah membuka atau mengedit gambar,
while True:
    pilihan = input("Pilih menu (1-5): ")
    if pilihan == "1":
        # Membuka gambar
        path = input("Masukkan path gambar: ")
        image = open_image(path)
        if image:
            print(f"Gambar dibuka: {path}")
            show_image(image)
        else:
```

Gambar 3.7 Membuka gambar

Source code :

```
image = None
# Di dalam loop, setelah membuka atau mengedit gambar, tambahkan pemanggilan fungsi
show_image
while True:
    pilihan = input("Pilih menu (1-5): ")
    if pilihan == "1":
        # Membuka gambar
        path = input("Masukkan path gambar: ")
        image = open_image(path)
        if image:
            print(f"Gambar dibuka: {path}")
            show_image(image)
        else:
            continue
```

Penjelasan kode :

image = None: Inisialisasi variabel image dengan None. Ini menandakan bahwa pada awalnya tidak ada gambar yang dibuka. **while True::** Membuat loop tak terbatas, yang berarti program akan terus berjalan sampai ada suatu kondisi untuk menghentikannya. **pilihan = input("Pilih menu (1-5): ")**: Mengambil input dari pengguna berupa pilihan menu,

yang disimpan dalam variabel pilihan. `if pilihan == "1":` Menggunakan kondisi if untuk mengecek apakah pengguna memilih menu nomor 1, yaitu "Membuka gambar".

Di dalam blok `if pilihan == "1":`, terdapat kode untuk membaca path gambar dari pengguna (`path = input("Masukkan path gambar: ")`) dan kemudian memanggil fungsi `open_image(path)` untuk membuka gambar. Jika gambar berhasil dibuka (nilai yang tidak None), program mencetak pesan bahwa gambar telah dibuka dan memanggil fungsi `show_image(image)` untuk menampilkan gambar tersebut. Jika gambar tidak berhasil dibuka, program menggunakan `continue` untuk melanjutkan ke iterasi berikutnya dari loop, meminta pengguna untuk memilih menu kembali.

H. Kode memotong gambar

```
elif pilihan == "2":
    # Memotong gambar
    if image:
        top = int(input("Masukkan batas atas: "))
        left = int(input("Masukkan batas kiri: "))
        bottom = int(input("Masukkan batas bawah: "))
        right = int(input("Masukkan batas kanan: "))
```

Gambar 3.8 Kode memotong gambar

Source code :

```
elif pilihan == "2":
    # Memotong gambar
    if image:
        top = int(input("Masukkan batas atas: "))
        left = int(input("Masukkan batas kiri: "))
        bottom = int(input("Masukkan batas bawah: "))
        right = int(input("Masukkan batas kanan: "))
```

Penjelasan kode :

`elif pilihan == "2":` Ini adalah bagian dari struktur pengkondisian yang mengecek apakah input pengguna (pilihan) sama dengan string "2". Jika ya, ini berarti pengguna memilih opsi untuk "Memotong gambar". `# Memotong gambar:` Ini adalah komentar yang memberikan deskripsi singkat tentang apa yang dilakukan oleh blok kode berikutnya. Dalam bahasa Inggris, artinya "Memotong gambar." `if image:` Kondisi ini memeriksa apakah variabel image telah diassign sebuah nilai selain None (yaitu, apakah suatu gambar telah dibuka atau dimuat). Jika image tidak None, ini berarti ada gambar untuk diproses, dan blok kode di dalam if akan dieksekusi.

Di dalam blok `if image:`, kode meminta pengguna untuk memasukkan empat nilai yang mewakili batas pemotongan:

top: Batas atas area pemotongan.

left: Batas kiri area pemotongan.

bottom: Batas bawah area pemotongan.

right: Batas kanan area pemotongan.

Nilai-nilai ini diperoleh menggunakan `input()` dan diubah menjadi bilangan bulat menggunakan `int(input(...))`.

I. Kode untuk Verifikasi

```
if top < bottom and left < right and 0 <= top < image.height and 0 <= left < image.width:
    cropped_image = crop_image(image, top, left, bottom, right)
    if cropped_image:
        print("Gambar berhasil dipotong!")
        image = cropped_image
        show_image(image)
    else:
        continue
else:
    print("Batas koordinat tidak valid!")
    continue
else:
    print("Buka gambar terlebih dahulu!")
```

Gambar 3.9 Kode verifikasi

Source code :

```
if top < bottom and left < right and 0 <= top < image.height and 0 <= left < image.width:
    cropped_image = crop_image(image, top, left, bottom, right)
    if cropped_image:
        print("Gambar berhasil dipotong!")
        image = cropped_image
        show_image(image)
    else:
        continue
else:
    print("Batas koordinat tidak valid!")
    continue
else:
    print("Buka gambar terlebih dahulu!")
```

Penjelasan kode :

`if top < bottom and left < right and 0 <= top < image.height and 0 <= left < image.width:` adalah kondisi yang memeriksa apakah batas koordinat yang dimasukkan

oleh pengguna valid. Kondisi ini memastikan bahwa top kurang dari bottom, left kurang dari right, dan bahwa nilai top dan left berada dalam rentang yang valid sesuai dengan ukuran gambar (image.height dan image.width). Di dalam blok `if`, jika batas koordinat valid, maka dilakukan pemotongan gambar dengan memanggil fungsi `crop_image(image, top, left, bottom, right)`. Hasilnya disimpan dalam variabel `cropped_image`.

`if cropped_image:` Kondisi ini memeriksa apakah operasi pemotongan gambar berhasil, yaitu `cropped_image` tidak `None`. Jika berhasil, program mencetak pesan "Gambar berhasil dipotong!", mengupdate variabel `image` dengan gambar yang telah dipotong, dan menampilkan gambar yang baru.

Jika operasi pemotongan gambar tidak berhasil (nilai `cropped_image` adalah `None`), maka program menggunakan `continue` untuk melanjutkan ke iterasi berikutnya dari loop. Jika batas koordinat tidak valid, program mencetak pesan "Batas koordinat tidak valid!" dan menggunakan `continue` untuk melanjutkan ke iterasi berikutnya dari loop. Jika tidak ada gambar yang dibuka sebelumnya (`image` adalah `None`), program mencetak pesan "Buka gambar terlebih dahulu!" dan menggunakan `continue` untuk melanjutkan ke iterasi berikutnya dari loop.

J. Kode untuk membuka gambar

```
elif pilihan == "3":
    # Mengubah ukuran gambar
    if image:
        width = int(input("Masukkan lebar baru: "))
        height = int(input("Masukkan tinggi baru: "))
        resized_image = resize_image(image, width, height)
        if resized_image:
            print("Gambar berhasil diubah ukuran!")
            image = resized_image
            show_image(image)
        else:
            continue
    else:
        print("Buka gambar terlebih dahulu!")

elif pilihan == "4":
```

Gambar 3.10 Kode untuk membuka gambar

Source code :

```
elif pilihan == "3":
    # Mengubah ukuran gambar
    if image:
        width = int(input("Masukkan lebar baru: "))
        height = int(input("Masukkan tinggi baru: "))
        resized_image = resize_image(image, width, height)
        if resized_image:
```

```

print("Gambar berhasil diubah ukuran!")
image = resized_image
show_image(image)
else:
continue
else:
print("Buka gambar terlebih dahulu!")

```

Penjelasan Kode :

`elif pilihan == "3":` Ini adalah bagian dari struktur pengkondisian yang mengecek apakah input pengguna (pilihan) sama dengan string "3". Jika ya, ini berarti pengguna memilih opsi untuk "Mengubah ukuran gambar". `if image:` Kondisi ini memeriksa apakah variabel `image` sudah di-assign nilai (yaitu, apakah gambar sudah dibuka atau dimuat). Jika ya, blok kode di dalamnya akan dieksekusi. Di dalam blok `if image:`, program meminta pengguna untuk memasukkan lebar dan tinggi baru untuk gambar, menggunakan `input()` untuk mengambil nilai lebar dan tinggi baru, dan mengonversinya menjadi bilangan bulat menggunakan `int(input(...))`.

`resized_image = resize_image(image, width, height):` Program memanggil fungsi `resize_image` dengan parameter gambar (`image`), lebar baru, dan tinggi baru. Hasilnya disimpan dalam variabel `resized_image`. `if resized_image:` Kondisi ini memeriksa apakah operasi perubahan ukuran gambar berhasil, yaitu apakah `resized_image` tidak `None`. Jika berhasil, program mencetak pesan "**Gambar berhasil diubah ukuran!**", mengganti variabel `image` dengan gambar yang telah diubah ukurannya, dan menampilkan gambar yang baru. Jika operasi perubahan ukuran gambar tidak berhasil (nilai `resized_image` adalah `None`), program menggunakan `continue` untuk melanjutkan ke iterasi berikutnya dari loop. Jika tidak ada gambar yang dibuka sebelumnya (`image` adalah `None`), program mencetak pesan "**Buka gambar terlebih dahulu!**" dan menggunakan `continue` untuk melanjutkan ke iterasi berikutnya dari loop.

K. Kode menyimpan gambar

```

elif pilihan == "4":
    # Menyimpan gambar
    if image:
        path = input("Masukkan path untuk menyimpan: ")
        save_image(image, path)
        print("Gambar berhasil disimpan!")
    else:
        print("Buka gambar terlebih dahulu!")

```

Gambar 3.11 Kode untuk menyimpan gambar

Source code :

```

elif pilihan == "4":
    # Menyimpan gambar
    if image:
        path = input("Masukkan path untuk menyimpan: ")
        save_image(image, path)
        print("Gambar berhasil disimpan!")
    else:
        print("Buka gambar terlebih dahulu!")

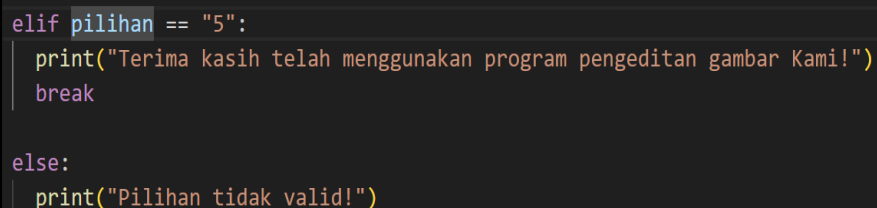
```

Penjelasan kode :

`elif pilihan == "4":` Ini adalah bagian dari struktur pengkondisian yang mengecek apakah input pengguna (pilihan) sama dengan string "4". Jika ya, ini berarti pengguna memilih opsi untuk "Menyimpan gambar". `# Menyimpan gambar`: Ini adalah komentar yang memberikan deskripsi singkat tentang apa yang dilakukan oleh blok kode berikutnya. Dalam bahasa Inggris, artinya "Save the image." `if image:` Kondisi ini memeriksa apakah variabel `image` sudah di-assign nilai (yaitu, apakah gambar sudah dibuka atau dimuat). Jika ya, blok kode di dalamnya akan dieksekusi. Di dalam blok `if image:`, program meminta pengguna untuk memasukkan path (lokasi) untuk menyimpan gambar yang sedang diolah, menggunakan `input()` untuk mengambil nilai path.

`save_image(image, path)`: Program memanggil fungsi `save_image` dengan parameter gambar (`image`) dan path untuk menyimpan (`path`). Fungsi ini mencoba menyimpan gambar ke lokasi yang ditentukan. Setelah menyimpan gambar, program mencetak pesan "Gambar berhasil disimpan!". Jika tidak ada gambar yang dibuka sebelumnya (`image` adalah `None`), program mencetak pesan "Buka gambar terlebih dahulu!"

L. Kode untuk pilihan 5 (keluar)



```

elif pilihan == "5":
    print("Terima kasih telah menggunakan program pengeditan gambar Kami!")
    break

else:
    print("Pilihan tidak valid!")

```

Gambar 3.12 Kode untuk pilihan 5 (Keluar)

Source code :

```

elif pilihan == "5":
    print("Terima kasih telah menggunakan program pengeditan gambar Kami!")
    break
else:

```

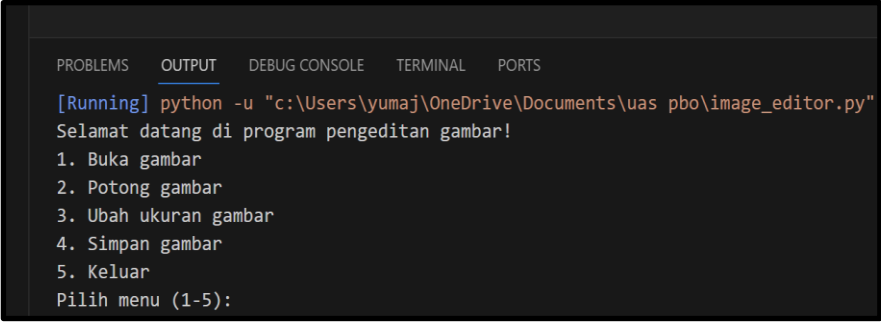
```
print("Pilihan tidak valid!")
```

Penjelasan kode:

`elif pilihan == "5":` Ini adalah bagian dari struktur pengkondisian yang mengecek apakah input pengguna (pilihan) sama dengan string "5". Jika ya, ini berarti pengguna memilih opsi untuk keluar dari program. Di dalam blok ini, program mencetak pesan "**Terima kasih telah menggunakan program pengeditan gambar Kami!**" untuk memberikan umpan balik kepada pengguna bahwa mereka telah keluar dari program. Selanjutnya, pernyataan `break` digunakan untuk menghentikan loop tak terbatas (`while True`) dan keluar dari program.

Jika pilihan pengguna tidak sama dengan "5", maka eksekusi akan masuk ke bagian `else`: Di bagian `else`, program mencetak pesan "**Pilihan tidak valid!**" untuk memberi tahu pengguna bahwa pilihan yang dimasukkan tidak sesuai dengan opsi yang valid. Setelah mencetak pesan ini, program akan kembali ke awal loop dan meminta pengguna untuk memilih kembali.

M. Output (luaran) dari Program



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
[Running] python -u "c:\Users\yumaj\OneDrive\Documents\uas pbo\image_editor.py"
Selamat datang di program pengeditan gambar!
1. Buka gambar
2. Potong gambar
3. Ubah ukuran gambar
4. Simpan gambar
5. Keluar
Pilih menu (1-5):
```

Gambar 3.13 Output Program

Source code :

Selamat datang di program pengeditan gambar!

1. Buka gambar
2. Potong gambar
3. Ubah ukuran gambar
4. Simpan gambar
5. Keluar

Pilih menu (1-5):

Penjelasan :

Buka gambar (Menu 1): Pengguna dapat memilih opsi ini untuk membuka sebuah gambar. Setelah memilih opsi ini, program akan meminta pengguna untuk memasukkan path (lokasi) gambar yang ingin dibuka.

Potong gambar (Menu 2): Jika pengguna memilih opsi ini, program akan memeriksa apakah sudah ada gambar yang dibuka sebelumnya. Jika ya, program akan meminta

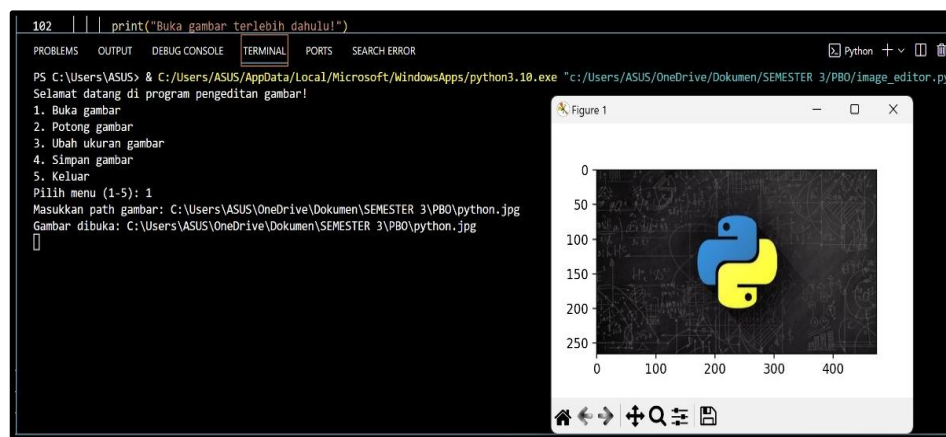
pengguna untuk memasukkan batas-batas pemotongan gambar, seperti batas atas, kiri, bawah, dan kanan. Setelah itu, program akan memanggil fungsi `crop_image` (yang harus didefinisikan di dalam kode program) untuk melakukan pemotongan gambar sesuai dengan batas yang dimasukkan.

Ubah ukuran gambar (Menu 3): Opsi ini pengguna mengubah ukuran gambar. Jika sudah ada gambar yang dibuka, program akan meminta pengguna untuk memasukkan lebar dan tinggi baru untuk gambar. Setelah itu, program akan memanggil fungsi `resize_image` (juga harus didefinisikan di dalam kode program) untuk mengubah ukuran gambar.

Simpan gambar (Menu 4): Jika pengguna memilih opsi ini, program akan memeriksa apakah sudah ada gambar yang dibuka. Jika ya, program akan meminta pengguna untuk memasukkan path (lokasi) di mana gambar akan disimpan. Kemudian, program akan memanggil fungsi `save_image` (harus didefinisikan) untuk menyimpan gambar ke lokasi yang dimasukkan.

Keluar (Menu 5): Jika pengguna memilih opsi ini, program akan mencetak pesan selamat tinggal dan keluar dari program.

N. Memilih menu 1

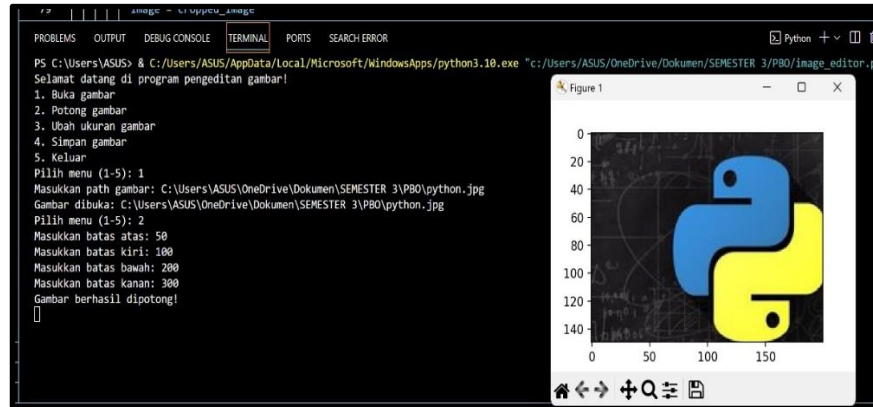


Gambar 3.14 membuka gambar

Penjelasan :

Pada gambar di atas merupakan penggunaan untuk menu 1 yaitu pengguna di minta untuk memasukkan angka 1-5 kemudian pengguna memasukkan angka 1 dan pengguna di minta untuk memasukkan link Alamat file gambar yang akan di buka setelah kemudian jika sudah di masukkan maka system akan membuka gambar yang kita masukkan tadi jika gambara berhasil dibuka maka program akan mencetak “gambar di buka” yang menandakan bahwa gambar berhasil di buka pada pilihan satu ini program hannya bisa membuka gambar saja dan pilihan satu ini merupakan induk dari semua pilihan yang dimana jika pilihan satu belum kita masukkan maka kita tidak akan bisa mengeksekusi atau mengakses pilihan pilihan lainnya.

O. Memotong gambar

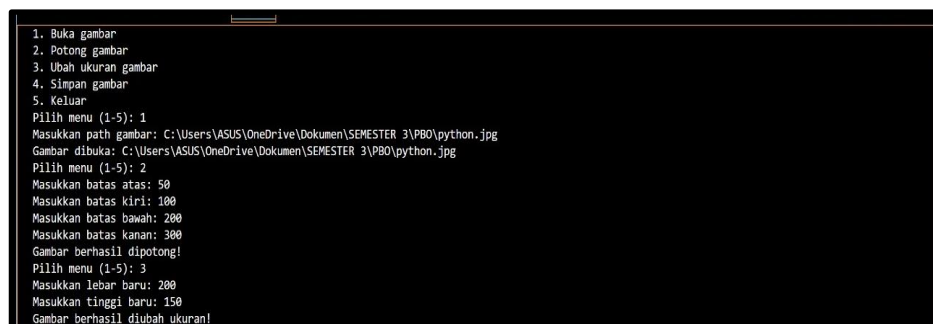


Gambar 3.15 memotong gambar

Penjelasan :

Pada menu kedua ini di gunakan untuk memotong gambar dimana pengguna diminta untuk memasukkan pilihannya yaitu memasukkan angka dua maka program akan meminta pengguna untuk memasukkan ukuran yang akan di potong mulai dari masukkan batas atas : 50, masukkan batas kiri 100, masukkan batas bawah : 200, masukkan batas kanan 300 kemudian jika kita sudah memasukkan batas-batas dari semua yang diminta oleh program kemudian enter jika gambar berhasil di potong maka program akan mencetak pesan gambar berhasil di potong dan kita bisa melihat gambar yang di potong tersebut.

P. Mengubah ukuran gambar



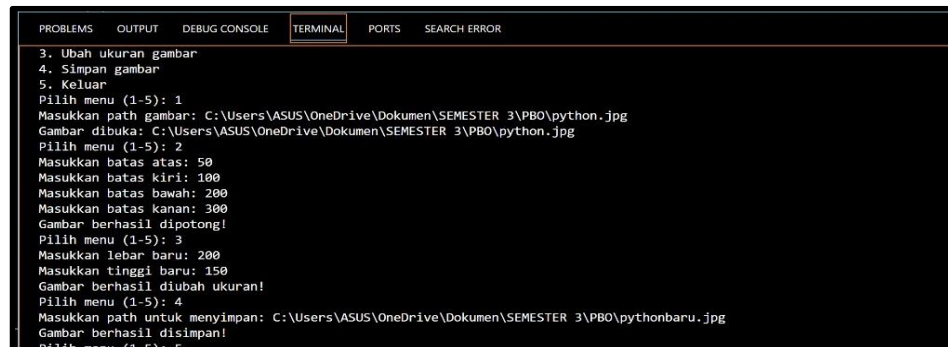
Gambar 3.16 mengubah ukuran gambar

Penjelasan :

Pada menu 3 ini merupakan menu untuk memotong gambar yaitu dimana pengguna di minta untuk memasukkan pilihan dan pengguna memasukkan pilihan tiga yaitu mengubah ukuran gambar kemudian enter maka pengguna di perintahkan untuk memasukkan ukuran baru sesuai dengan keinginan pengguna yaitu pengguna diminta untuk memasukkan lebar baru : 200, masukkan tinggi baru :150 kemudian enter jika

gambar berhasil di ubah maka program akan mencetak pesan gambar berhasil di ubah ukuran.

Q. Simpan gambar



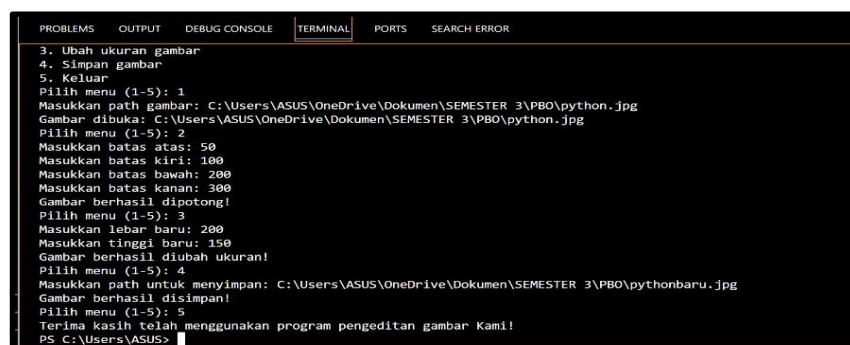
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
3. Ubah ukuran gambar
4. Simpan gambar
5. Keluar
Pilih menu (1-5): 1
Masukkan path gambar: C:\Users\ASUS\OneDrive\Dokumen\SEMESTER 3\PBO\python.jpg
Gambar dibuka: C:\Users\ASUS\OneDrive\Dokumen\SEMESTER 3\PBO\python.jpg
Pilih menu (1-5): 2
Masukkan batas atas: 50
Masukkan batas kiri: 100
Masukkan batas bawah: 200
Masukkan batas kanan: 300
Gambar berhasil dipotong!
Pilih menu (1-5): 3
Masukkan lebar baru: 200
Masukkan tinggi baru: 150
Gambar berhasil diubah ukuran!
Pilih menu (1-5): 4
Masukkan path untuk menyimpan: C:\Users\ASUS\OneDrive\Dokumen\SEMESTER 3\PBO\pythonbaru.jpg
Gambar berhasil disimpan!
Pilih menu (1-5): 5
```

Gambar 3.17 menyimpan gambar

Penjelasan :

Pada menu ini pengguna diminta untuk memasukkan pilihan mana yang ingin di jalankan dan pengguna memasukkan pilihan 4 yaitu program untuk menyimpan gambar kemudian enter maka program akan meminta pengguna untuk meletakkan atau mengisi Alamat lokasi untuk menyimpan gambar tersebut setelah di isi kemudian enter jika gambar berhasil di simpan maka program akan mencetak pesan yaitu gambar berhasil di simpan yang berarti gambar tersebut sudah di simpan sesuai dengan lokasi yang kita masukkan tadi .

R. Keluar



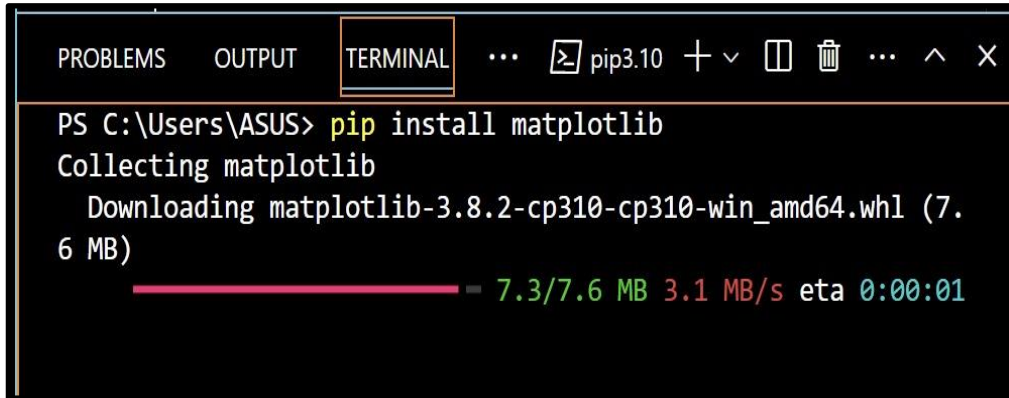
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
3. Ubah ukuran gambar
4. Simpan gambar
5. Keluar
Pilih menu (1-5): 1
Masukkan path gambar: C:\Users\ASUS\OneDrive\Dokumen\SEMESTER 3\PBO\python.jpg
Gambar dibuka: C:\Users\ASUS\OneDrive\Dokumen\SEMESTER 3\PBO\python.jpg
Pilih menu (1-5): 2
Masukkan batas atas: 50
Masukkan batas kiri: 100
Masukkan batas bawah: 200
Masukkan batas kanan: 300
Gambar berhasil dipotong!
Pilih menu (1-5): 3
Masukkan lebar baru: 200
Masukkan tinggi baru: 150
Gambar berhasil diubah ukuran!
Pilih menu (1-5): 4
Masukkan path untuk menyimpan: C:\Users\ASUS\OneDrive\Dokumen\SEMESTER 3\PBO\pythonbaru.jpg
Gambar berhasil disimpan!
Pilih menu (1-5): 5
Terima kasih telah menggunakan program pengeditan gambar Kami!
PS C:\Users\ASUS>
```

Gambar 3.18 Keluar

Penjelasan :

Pada gambar di atas merupakan pilihan untuk keluar dimana jika pengguna ingin keluar dari program ini pengguna dapat memilih pilihan lima yaitu keluar dan program akan mencetak pesan terima kasih telah menggunakan program pengeditan gambar kami!.

S. Install matplotlib



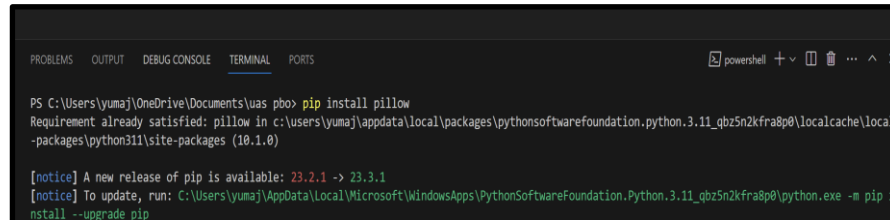
```
PROBLEMS OUTPUT TERMINAL ... pip3.10 + v [ ] [ ] ... ^ X
PS C:\Users\ASUS> pip install matplotlib
Collecting matplotlib
  Downloading matplotlib-3.8.2-cp310-cp310-win_amd64.whl (7.6 MB)
    = 7.3/7.6 MB 3.1 MB/s eta 0:00:01
```

Gambar 3.19 Install matplotlib

Penjelasan :

Untuk menjalankan program pengeditan gambar ini kita perlu menginstal matplotlib dan pillow dimana pada gambar di atas merupakan penginstallan matplotlib adalah salah satu perpustakaan (library) dalam bahasa pemrograman Python yang digunakan untuk membuat visualisasi grafik dan plot. Penginstallan Matplotlib umumnya dilakukan untuk keperluan visualisasi data, analisis data, dan representasi grafis dari hasil perhitungan atau eksplorasi data.

T. Instal pillow



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS ... powershell + v [ ] [ ] ... ^ X
PS C:\Users\yumaj\OneDrive\Documents\uas pbox> pip install pillow
Requirement already satisfied: pillow in c:\users\yumaj\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (10.1.0)

[notice] A new release of pip is available: 23.2.1 -> 23.3.1
[notice] To update, run: C:\Users\yumaj\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\python.exe -m pip install --upgrade pip
```

Gambar 3.20 Install pillow

Penjelasan :

Pillow adalah library Python yang digunakan untuk pemrosesan gambar. Fungsi utama dari Pillow adalah memungkinkan Anda membuka, memanipulasi, dan menyimpan berbagai format file gambar. Beberapa tugas umum yang dapat dilakukan dengan menggunakan Pillow Membuka dan Menyimpan Gambar: Pillow mendukung berbagai format file gambar seperti JPEG, PNG, GIF, BMP, dan lainnya. Anda dapat menggunakan Pillow untuk membuka gambar dari file dan menyimpannya ke format yang berbeda. Anda dapat menggunakan Pillow untuk melakukan berbagai operasi manipulasi gambar seperti memutar, memotong, memperkecil, atau memperbesar gambar. Pillow mendukung konversi mode warna gambar, yang berguna jika Anda perlu mengubah gambar ke skala warna yang berbeda. Pillow juga menyediakan berbagai fungsi lain seperti penambahan teks ke gambar, rotasi, manipulasi histogram, dan operasi lainnya.

BAB IV

KESIMPULAN DAN SARAN

4.1 KESIMPULAN

Program ini merupakan dasar dari sebuah program pengeditan gambar sederhana dengan antarmuka teks. Program ini memberikan pengguna beberapa opsi, seperti membuka, memotong, mengubah ukuran, dan menyimpan gambar. Tujuan utamanya adalah memberikan kemudahan kepada pengguna untuk melakukan operasi dasar pengeditan gambar tanpa memerlukan antarmuka grafis dan pengguna bisa menggunakan opsi opsi tersebut dengan cara memasukkan pilihan mana yang akan si buka Kesimpulannya, program ini menyajikan antarmuka yang ramah pengguna untuk melakukan manipulasi dasar pada gambar secara interaktif melalui input teks.

4.2 SARAN

Pastikan pemahaman yang baik tentang konsep dasar OOP, seperti kelas, objek, pewarisan, enkapsulasi, dan polimorfisme. Ini membantu dalam merancang dan mengimplementasikan struktur program yang efektif. Pisahkan fungsionalitas ke dalam kelas dan modul yang terpisah. Ini meningkatkan keterbacaan dan mempermudah pemeliharaan. Gunakan Pewarisan dengan Bijak: Gunakan pewarisan untuk merinci hubungan "is-a" antar kelas. Hindari pewarisan yang berlebihan dan pertimbangkan penggunaan komposisi jika lebih sesuai.

Implementasikan uji coba unit untuk memastikan setiap kelas dan metode berfungsi sesuai harapan. Uji coba otomatis (automated testing) dapat membantu mendeteksi kesalahan lebih awal. Hindari menggunakan variabel global yang dapat mengakibatkan kesulitan dalam pemeliharaan dan memahami alur program. Gunakan objek dan metode untuk membatasi ruang lingkup. Pelajari Library dan Framework OOP: Manfaatkan library dan framework Python yang sudah mendukung paradigma OOP, seperti Django untuk pengembangan web atau Pygame untuk pembuatan game.

DAFTAR PUSTAKA

- Angkati Permani (2019), *“Pengembangan Modul Elektronik Pemrograman Berorientasi Objek untuk Siswa Kelas XI Rekayasa Perangkat Lunak dengan Model Four-D”* <https://journal.uny.ac.id/index.php/elinvo/article/view/28260> (diakses pada 13 Desember 2023).
- Diandra Lestriani H (2019) *“Penggunaan Bahasa Pemrograman Python Dalam Analisis Faktor Penyebab Kanker Paru-Paru”* <https://ejurnal.stie trianandra.ac .id/index.php/jupti/ article/ view/1742> (Diakses pada 13 desember 2023).
- Muhammad Siddik (2019). *“Pengembangan Sistem Informasi Administrasi Akademik Dengan Rancangan Modul Program Menggunakan Bahasa Pemrograman Berorientasi Objek”*<https://www.ejournal. pelitaindonesia .ac.id /ojs32 /index.php/JOISIE/article /view/251> (Diakses pada 13 Desember 2023).
- Yuliawati Yunus (2019) *“VALIDITAS MEDIA E-MODUL PEMOGRAMAN BERORIENTASI OBJEK II BERBASIS PROBLEM BASED INSTRUCTION”* <http://ejournal.ildikti10.id/index.php/curricula/article/view/3959> (diakses pada 13 desember 2023).