Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	**71230982**
Nama Lengkap	**Rainie Fanita Chrisabel Hadisantoso**
Minggu ke / Materi	**12 / Tipe Data Set**

SAYA MENYATAKAN BAHWA LAPORAN PRAKTIKUM INI SAYA BUAT DENGAN USAHA SENDIRI TANPA MENGGUNAKAN BANTUAN ORANG LAIN. SEMUA MATERI YANG SAYA AMBIL DARI SUMBER LAIN SUDAH SAYA CANTUMKAN SUMBERNYA DAN TELAH SAYA TULIS ULANG DENGAN BAHASA SAYA SENDIRI.

SAYA SANGGUP MENERIMA SANKSI JIKA MELAKUKAN KEGIATAN PLAGIASI, TERMASUK SANKSI TIDAK LULUS MATA KULIAH INI.

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2024

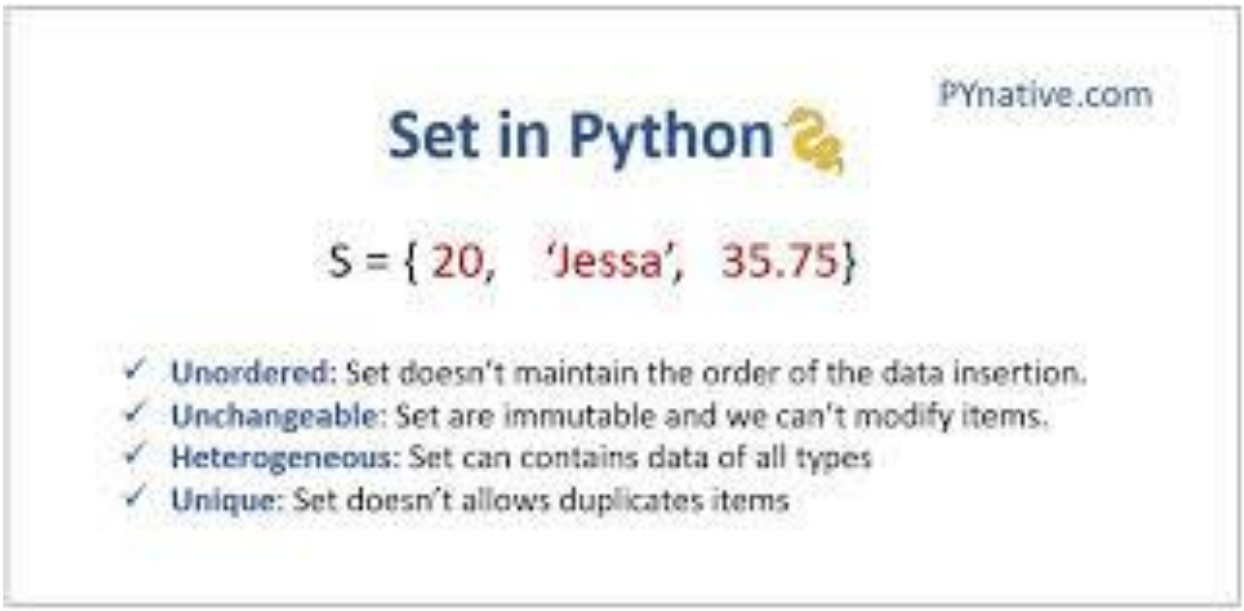
MATERI : Percabangan dan Perulangan Kompleks

Dalam pembelajaran ke 13 dengan materi bab 12: Percabangan dan Perulangan Kompleks, ada beberapa tujuan yang hendak dicapai:

1. Melakukan operasi-operasi dasar pada tipe data Set. Operasi-operasi yang dimaksud adalah seperti mendefinisikan, mengakses, mengubah, menghapus dan melakukan pencarian data pada Set
2. Melakukan operasi-operasi yang melibatkan dua set atau lebih, seperti union, difference, intersection dan symmetric difference

Pengenalan dan Mendefinisikan Set

Python memiliki tipe data bawaan yang unik hanya ada di Python, 4 build-in tipe data Python tersebut adalah List, tuple, Dictionary dan Set. Dala pembelajaran ini akan dibahas tipa data set. Tipe data set berfungsi untuk menampung sekumpulan data yang bersifat unik. Tipe data set juga kalau dalam dunia matmatika disebut dengan himpunan.



Data-data dalam set memiliki karakteristik:

1. Unindexed

   Berbeda dengan tipe data string dan list, anggota-anggota dalam set tidak memiliki indeks.

2. Unordered :

   Karena anggota-anggota dalam set tidak diberikan indeks, urutan data tidak terpatok pada urutan data yang disajikan maupun urutan data dimasukkan kedalam set. Sehingga urutan anggota-anggota dalam set dapat berubah-ubah ketika diproses maupun ditampilkan.

3. Unchangeable:

Anggota-anggota dalam set bersifat immutable atau tidak dapat diubah-ubah. Namun tipe data set itu sendiri dapat berubah-ubah.

4. Heterogeneous:

   Dapat menyimpan berbagai jenis tipe data di dalamnya. Dengan syarat bahwa tipe datanya bersifat immutable.

5. Unique:

   Anggota-anggota dalam sebuah himpunan set harus berbeda. Ketika ada anggota dengan nilai yang sama, maka hanya salah satunya yang dianggap dan ditampilkan.

```
angka = {1,2,3,4,5,5}
print(angka)    #{1, 2, 3, 4, 5}
```

   Begitu juga hanya dengan True = 1 dan False = 0. Kalau dalam khasus mereka, urutan dalam himpunan set berpengaruh dalam bentuk penampilan datanya. Misalnya urutannya 1 ada didepan True, maka 1 yang akan diambil, begitu juga sebaliknya, jika True lebih dulu dari 1, maka yang ditampilkan adalah True.

```
trufal = {1, True, False, 0}
print(trufal)   #{False, 1}
```

Berikut beberapa sifat set dalam Python:

- Seperti dalam penyebutan himpunan, data-data dalam set disebut dengan anggota (member)
- Anggota- anggota dalam sebuah set harus bersifat immutable, contohnya tipe data interger, float, string, tuple, dan lainya. Karena list dan dictionary bersifat mutable, maka mereka tidak dapat dimasukkan sebagai anggota dalam sebuah set.
- Tipe data set sendiri juga besifat mutable. Kita dapat menambah dan mengurangi isi dalam sebuah set. Jadi pokoknya, isi set dapat diganti, tapi anggota dalam set tidak dapat diubah-ubah. Karena set itu mutable maka kita tidak dapat menambahkan tipe data set sebagia anggota dalam sebuah set.

Untuk membuat tipe data set kita dapatmenggunakan bebrapa cara, yaitu sebagai berikut:

1. Notasi {}

```
b_ganjil = {1, 3, 5, 7, 9, 11}
b_genap = {2, 4, 6, 8, 10, 12}

print(type(b_ganjil), type(b_genap))
#<class 'set'> <class 'set'>
```

2. Fungsi set()
   Langsung dengan fungsi set() agar data-data yang diberikan disimpandalam bentuk tipe data set.

```
nama = set(("rain", "amel", "jose", "ria"))
print(type(nama))   #<class 'set'>
```

Hal yang perlu sangat diperhatikan dalam menggunakan fungsi set() untuk membuat tipe data set adalah harus menggunakan dalam kurung lagi untuk mengemas karakter-karakter didalamnya. Jika tidak menggunakan tabda kurung dua kali, maka akan terjadi error. Hal tersebut karena, set hanya meminta sebuah kumpulan data, jika data-data tidak dijadikan satu dalam sebuah tanda kurung, maka akan dianggap sebagai banyak data.

```
nama = set("rain", "amel", "jose", "ria")
print(type(nama))   #<class 'set'>

#nama = set("rain", "amel", "jose", "ria")
#       ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
#TypeError: set expected at most 1 argument, got 4
```

Kalau ingin membuat sebuah variabel tipe data set kosong juga harus meggunakan fungsi set(. Jika menggunakan notasi {} maka variabel akan dianggap sebagai dictionary dan bukan tipe data set.

```
iniset = set()
print(type(iniset)) #<class 'set'>

notset = {}
print(type(notset)) #<class 'dict'>
```

Pengaksesan Set

Beberapa karakteristik yang dimiliki oleh set adalah unindexed (tidak memiliki indeks seperti list atau string unordered (tidak urut), unchangeable (tidak dapat diubah). Karena set unindexed atau tidak memiliki indeks, maka data-data dalam set tidak dapat diakses degan langsing menggunakan indeks.

```
nim = {"71230982", "71230976", "71231005", "71230998"}
jumlah_nim = len(nim)
print(jumlah_nim)       #4

for n in nim:
    print(n)
```

Output dari kode diatas sebagai berikut:

```
4
71230982
71230976
71231005
71230998
```

Dari output program diatas, dapat dilihat bahwa urutan anggota nim yang ditampilkan berbeda dengan urutan anggota-anggota set yang dideklarasikan dalam program. Hal tersebut terjadi karena mereka tidak memiliki indeks, sehingga setiap kali data-data terseut disimpan dalam memory, data tidak dimasukkan secara urut dan tetap. Jika sebuah tipe data memiliki indeks, maka urutan memauskkan data dalam memory pun juga urut sesuai dengan indeks yang dimiliki data tersebut. Sehingga tipe data set sangat disarankan untuk digunakan ketika pengolahan angggota-anggota didalamnya tidak membutuhkan urutan yang exact atau urutan bukan sebuah kepentingan dalam pengolahannya(bukan mengurutkan data secara ascending dan descending.

Meskipun anggota-anggota dalam set harus bersifat immutable, set sendiri adalah tipe data yang bersifat mutable. Jadi anggota dalam tipe data set dapat ditambah maupun dikurangi. Untuk menambah data kedalam sebuah himpunan set, dapat digunakan fungsi add(), sedangkan untuk mengurangi data dapat digunakan fungsi remove(), discard(), pop() maupun clear().

- **Menambahkan anggota:**
  Contoh program menambahkan nama minuman dalam sebuah set minuman.

```
minuman = set()

minuman.add("Hot Choco")
minuman.add("Macha Latte")
minuman.add("Caramel Machiatto")

print(len(minuman))

minuman.add("Macha Latte")

for minum in minuman:
    print(minum)
```

  Output dari program diatas adalah

```
3
Macha Latte
Caramel Machiatto
Hot Choco
```

  Dapat dilihat bahwa meskipun “Macha Latte” dimasukkan dua kali, namun tetap saja yang ditampilkan hanya satu “Macha Latte” saja. Hal tersebut terjadi karena setiap ada anggota baru, set akan mengcek apakah anggotanya sudah ada didalamnya atau tidak. Jika belum ada maka anggota baru akan ditambahkan, namun jika sudah ada maka anggota tidak akan ditambahkan.

- **Menghapus anggota**

  Berikut beberapa cara yang dapat digunakan untuk menghapus satu atau lebih anggota dalam set.

discard()	remove()	pop()	clear()
Menghapus satu elemen yang disebutkan	Menghapus satu elemen yang disebutkan	Mengambil salah satu dan menghapusnya dari set (tidak tentu)	Menghapus seluruh elemen di dalam set
Tidak ada error	Muncul error jika elemen yang dihapus tidak ada	Error jika set kosong	Tidak ada error

Berikut diberikan sebuah data bertipe set berisikan bialngan-bialngan ganjil.

```
bilangan_prima = {2, 3, 5, 7, 11, 13, 23, 29}
print(bilangan_prima)
# {2, 3, 5, 7, 11, 13, 23, 29}
```

1. Remove(): Memhapus elemen dengan cara menyebutkan elemen yang ingin dihapus.

```
bilangan_prima.remove(5)
print(bilangan_prima)
# {2, 3, 7, 11, 13, 23, 29}
```

2. Discard(): Sama dengan remove() namun tidak ada pesan error jika elemen tidak dapat ditemukan

```
bilangan_prima.discard(97)
print(bilangan_prima)
# Karena tidak ada 97 dalam set maka tidak mempegaruhi
```

3. Pop():Mengambil salah satu anggota secara random dan menghapusnya dari dalam set.

```
bilangan = bilangan_prima.pop()
print(bilangan)
# 13
print(bilangan_prima)
# {2, 3, 7, 11, 23, 29}
```

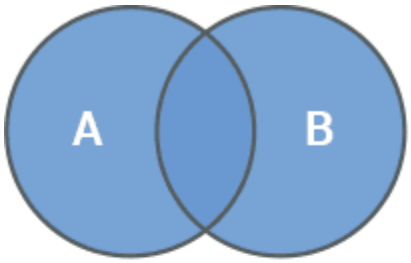
4. Clear(): Menghapus semua anggota dalam set.

```
bilangan_prima.clear()
print(bilangan_prima)
#None
```

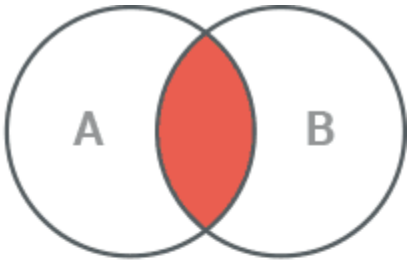
Maka jika ingin mengganti anggota dalam sebuah set, dapat dilakukan dengan cara menghapus anggota yang mau dihapus, kemudian menambah anggota baru secara terpisah.

Operasi-Operasi pada Set

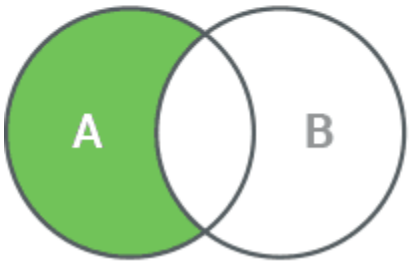
Dalam Python kita dapat mengolah set seperti mengolah himpunan. Untuk jenis-jenis operasi himpunan tentu seharusnya sudah dipelajari dipelajaran matematika. Berikut gambaran operasi-operassi yang dapat dilakukan:



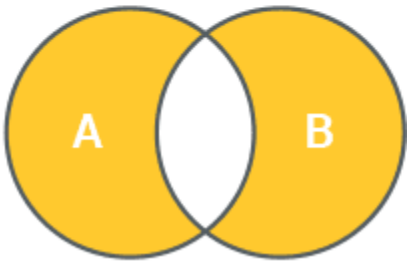
Union



Intersection



Difference



Symmetric Difference

Berikut tabel penjelasan beserta lambang-lambang notasi yang dapat digunakan:

Operation	Equivalent	Result
`len(s)`		number of elements in set *s* (cardinality)
`x in s`		test *x* for membership in *s*
`x not in s`		test *x* for non-membership in *s*
`s.issubset(t)`	`s <= t`	test whether every element in *s* is in *t*
`s.issuperset(t)`	`s >= t`	test whether every element in *t* is in *s*
`s.union(t)`	`s | t`	new set with elements from both *s* and *t*
`s.intersection(t)`	`s & t`	new set with elements common to *s* and *t*
`s.difference(t)`	`s - t`	new set with elements in *s* but not in *t*
`s.symmetric_difference(t)`	`s ^ t`	new set with elements in either *s* or *t* but not both
`s.copy()`		new set with a shallow copy of *s*

BAGIAN 2: LATIHAN MANDIRI (60%)

SOAL 1

Membuat program untuk memasukkan input dari user berupa jumlah kategori aplikasi, nama-nama kategori aplikasi beserta daftar aplikasi-aplikasinya. Untuk kategori aplikasi berjumlah 2, maka akan dicari aplikasi-aplikasi yang muncul di 1 kategori saja. Untuk kategori aplikasi berjumlah lebih dari dua, maka akan dicari aplikasi-aplikasi yang muncul di 2 kategori saja.

a. Source code

```
n = int(input('Masukkan jumlah kategori: '))

daftar_app = {}

for i in range(n):
    kate = input('Masukkan nama kategori:')
    print('Masukkan 5 nama aplikasi di kategori', kate)
    app = []
    for j in range(5):
        nama_app = input('Nama aplikasi: ')
        app.append(nama_app)
    daftar_app[kate] = app

daftar_app_ls = []

for app in daftar_app.values():
    daftar_app_ls.append(set(app))

if n == 2:
    sym_diff = daftar_app_ls[0].symmetric_difference(daftar_app_ls[1])
    print("Aplikasi Yang Hanya Muncul Di Satu Kategori Saja: ", sym_diff)
elif n > 2:
    for i in range(1,n):
        sym_diff = daftar_app_ls[i] and daftar_app_ls[i-1]
    else:
        sym_diff = daftar_app_ls[0] and daftar_app_ls[1]
    print("Aplikasi Yang Muncul Di Dua Kategori Sekaligus: ", sym_diff)
```

b. Output

- 2 Kategori:

```
PS C:\Users\asus\Downloads\File> python -u "c:\Users\asus\Downloads\File\Latihan_
1.py"
Masukkan jumlah kategori: 2
Masukkan nama kategori:a
Masukkan 5 nama aplikasi di kategori a
Nama aplikasi: 1
Nama aplikasi: 2
Nama aplikasi: 3
Nama aplikasi: 4
Nama aplikasi: 5
Masukkan nama kategori:b
Masukkan 5 nama aplikasi di kategori b
Nama aplikasi: 3
Nama aplikasi: 4
Nama aplikasi: 5
Nama aplikasi: 6
Nama aplikasi: 7
Aplikasi Yang Hanya Muncul Di Satu Kategori Saja:  {'6', '1', '7', '2'}
```

- 3 Kategori:

```
PS C:\Users\asus\Downloads\File> python -u "c:\Users\asus\Downloads\File\Latihan_12_7
1.py"
Masukkan jumlah kategori: 3
Masukkan nama kategori:a
Masukkan 5 nama aplikasi di kategori a
Nama aplikasi: 1
Nama aplikasi: 2
Nama aplikasi: a
Nama aplikasi: b
Nama aplikasi: c
Masukkan nama kategori:b
Masukkan 5 nama aplikasi di kategori b
Nama aplikasi: 1
Nama aplikasi: 2
Nama aplikasi: d
Nama aplikasi: e
Nama aplikasi: f
Masukkan nama kategori:c
Masukkan 5 nama aplikasi di kategori c
Nama aplikasi: 3
Nama aplikasi: 4
Nama aplikasi: d
Nama aplikasi: e
Nama aplikasi: f
Aplikasi Yang Muncul Di Dua Kategori Sekaligus:  {'e', 'd', '2', 'f', '1'}
```

c. Penjelasan

- Pengguna diminta untuk memasukkan jumlah kategori aplikasi menggunakan n = int(input('Masukkan jumlah kategori: ')).
- Untuk setiap kategori, pengguna diminta memasukkan nama kategori dan lima nama aplikasi dengan menggunakan loop for dan input().
- Nama kategori dan aplikasi disimpan dalam sebuah kamus dengan menggunakan pernyataan daftar_app[kate] = app.

- Setiap daftar aplikasi dikonversi menjadi set untuk menghilangkan duplikasi dengan menggunakan [set(app) for app in daftar_app.values()].
- Jika jumlah kategori adalah 2, program mencari perbedaan simetris antara dua set aplikasi dengan menggunakan sym_diff = daftar_app_ls[0].symmetric_difference(daftar_app_ls[1]).
- Jika jumlah kategori lebih dari 2, program mencoba menemukan aplikasi yang muncul di dua kategori sekaligus dengan menggunakan operasi set, namun logika di sini tidak memberikan hasil yang diinginkan.
- Hasil perbedaan aplikasi ditampilkan kepada pengguna dengan menggunakan print()

SOAL 2

Tulis jawaban anda untuk soal nomor 1 di sini. Hapus paragraf ini.

a. Source code

```
data = input("Masukkan data yang mau dikonversi: ")

print("List ke Set")
data = list(data)
print("List: ",data)
data = set(data)
print("Set: ",data)
print()

print("Set ke List")
print("Set: ",data)
data = list(data)
print("List: ",data)
print()

print("Tuple ke Set")
data = tuple(data)
print("Tuple: ",data)
data = set(data)
print("Set: ",data)
print()

print("Set ke Tuple")
data = set(data)
print("Set: ",data)
data = tuple(data)
print("Tuple: ",data)
```

b. Output

```
PS C:\Users\asus\Downloads\File> python -u "c:\Users\asus\Downloads\File\Latihan_12_71230982\Latihan_12_
2.py"
Masukkan data yang mau dikonversi: Praktikum Algoritma dan Perograman
List ke Set
List:  ['P', 'r', 'a', 'k', 't', 'i', 'k', 'u', 'm', ' ', 'A', 'l', 'g', 'o', 'r', 'i', 't', 'm', 'a', '
 ', 'd', 'a', 'n', ' ', 'P', 'e', 'r', 'o', 'g', 'r', 'a', 'm', 'a', 'n']
Set:  {' ', 'e', 'P', 'u', 'm', 'r', 'l', 'A', 't', 'o', 'g', 'd', 'i', 'n', 'a', 'k'}

Set ke List
Set:  {' ', 'e', 'P', 'u', 'm', 'r', 'l', 'A', 't', 'o', 'g', 'd', 'i', 'n', 'a', 'k'}
List:  [' ', 'e', 'P', 'u', 'm', 'r', 'l', 'A', 't', 'o', 'g', 'd', 'i', 'n', 'a', 'k']

Tuple ke Set
Tuple:  (' ', 'e', 'P', 'u', 'm', 'r', 'l', 'A', 't', 'o', 'g', 'd', 'i', 'n', 'a', 'k')
Set:  {' ', 'e', 'P', 'm', 'u', 'r', 'l', 'A', 't', 'o', 'g', 'd', 'i', 'n', 'a', 'k'}

Set ke Tuple
Set:  {'P', 'm', 'r', 'A', 't', 'o', 'g', 'i', 'n', 'a', ' ', 'e', 'u', 'l', 'd', 'k'}
Tuple:  ('P', 'm', 'r', 'A', 't', 'o', 'g', 'i', 'n', 'a', ' ', 'e', 'u', 'l', 'd', 'k')
```

c. Penjelasan

- Pengguna diminta memasukkan data yang akan dikonversi menjadi set menggunakan data = input("Masukkan data yang mau dikonversi: ").
- Data dimasukkan ke dalam list untuk persiapan konversi ke set:
  1. Data dimasukkan ke dalam list dengan data = list(data).
  2. List awal ditampilkan menggunakan print("List: ", data).
- List dikonversi menjadi set untuk menghilangkan duplikasi:
  1. List dikonversi menjadi set dengan data = set(data).
  2. Set hasil konversi ditampilkan menggunakan print("Set: ", data).
- Set dikonversi kembali menjadi list:
  1. Set dikonversi kembali menjadi list dengan data = list(data).
  2. List hasil konversi ditampilkan menggunakan print("List: ", data).
- Data dalam tuple diubah menjadi set:
  1. Tuple diubah menjadi set dengan data = set(data).
  2. Set hasil konversi ditampilkan menggunakan print("Set: ", data).
- Set dikonversi kembali menjadi tuple:
  1. Set dikonversi kembali menjadi tuple dengan data = tuple(data).
  2. Tuple hasil konversi ditampilkan menggunakan print("Tuple: ", data)

SOAL 3

Tulis jawaban anda untuk soal nomor 1 di sini. Hapus paragraf ini.

a. Source code

```
t1 = input("Masukkan nama file teks 1: ")
t2 = input("Masukkan nama file teks 2: ")

try:
    with open(t1, 'r') as f1, open(t2, 'r') as f2:
        kata1 = set(kata.lower() for baris in f1 for kata in baris.split())
```

```
        kata2 = set(kata.lower() for baris in f2 for kata in baris.split())

    sama = kata1.intersection(kata2)
    print("Kata-kata yang muncul pada kedua file:", sama)

except FileNotFoundError:
    print("File tidak ditemukan")
```

b. Output

```
PS C:\Users\asus\Downloads\File> python -u "c:\Users\asus\Downloads\File\Latihan_12_712309
3.py"
Masukkan nama file teks 1: teks1.txt
Masukkan nama file teks 2: teks2.txt
Kata-kata yang muncul pada kedua file: {'hadisantoso', 'chrisabel', 'rainie', 'fanita'}
```

c. Penjelasan

- Pengguna diminta untuk memasukkan nama dua file teks.
- Program mencoba membuka kedua file tersebut untuk dibaca. Ini dilakukan dengan menggunakan pernyataan with.
- Setiap kata dalam file teks pertama dan kedua dibaca dan diubah menjadi huruf kecil menggunakan list comprehension. Setiap kata dibagi menggunakan split() dan diubah menjadi huruf kecil menggunakan lower(). Kemudian, kata-kata ini dimasukkan ke dalam dua set terpisah, kata1 dan kata2.
- Operasi intersection digunakan antara dua set kata-kata (kata1 dan kata2) untuk menemukan kata-kata yang muncul pada kedua file.
- Kata-kata yang sama pada kedua file ditampilkan kepada pengguna dengan mencetak set kata-kata yang sama.
- Jika salah satu file tidak ditemukan, program menangani pengecualian FileNotFoundError dan mencetak pesan kesalahan yang sesuai.

Link Github:

https://github.com/rainiefch/Praktikum-Algoritma-Pemrograman_71230982/tree/f31ed927244b24af116e3645d93bdaf899f2d708/Latihan%2012