



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71230971
Nama Lengkap	James Marvin Santoso
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

BAGIAN 1: MATERI MINGGU INI

PENGENALAN DAN MENDEFINISIKAN SET

Set dalam Python adalah struktur data yang digunakan untuk menyimpan kumpulan elemen yang unik. Set tidak mengizinkan adanya elemen duplikat, sehingga setiap elemen dalam set hanya muncul sekali. Set dalam Python didefinisikan dengan menggunakan kurung kurawal {} dan elemen-elemen dalam set dipisahkan oleh koma. Set juga tidak mempertahankan urutan elemen, sehingga elemen-elemen tersebut tidak dapat diakses berdasarkan indeks. Set dalam Python sangat berguna ketika kita perlu menyimpan kumpulan elemen yang unik dan tidak memerlukan urutan tertentu.

Berikut adalah contoh penggunaan set dalam Python:

```
# Membuat sebuah set kosong
my_set = set()

# Membuat sebuah set dengan item-item tertentu
my_set = {1, 2, 3, 4, 5}

# Menampilkan set
print(my_set)
```

Pada contoh di atas, kita membuat sebuah set kosong dengan set() dan sebuah set dengan beberapa item. Ketika kita mencetak set, item-itemnya akan ditampilkan dalam urutan yang tidak terdefinisi, karena set tidak mengatur urutan item-itemnya.

Selain itu, set juga memiliki beberapa metode bawaan yang dapat digunakan untuk melakukan operasi seperti menambahkan item, menghapus item, atau melakukan operasi matematika seperti gabungan, irisan, dan selisih antara dua set.

Berikut adalah contoh penggunaan metode-metode ini:

```
# Menambahkan elemen ke dalam set
my_set.add(6)
print(my_set)

# Menghapus elemen dari set
my_set.remove(3)
print(my_set)

# Melakukan iterasi pada set
for item in my_set:
    print(item)

# Menggunakan operasi matematika pada set
```

```
set1 = {1, 2, 3, 4, 5}
set2 = {4, 5, 6, 7, 8}

# Gabungan dua set
union_set = set1.union(set2)
print(union_set)

# Irisan dua set
intersection_set = set1.intersection(set2)
print(intersection_set)

# Selisih dua set
difference_set = set1.difference(set2)
print(difference_set)
```

Dalam contoh ini, kita menambahkan elemen baru ke dalam set dengan `add()`, menghapus elemen dari set dengan `remove()`, melakukan iterasi pada set dengan loop `for`, dan menggunakan beberapa metode set lainnya untuk melakukan operasi seperti gabungan, irisan, dan selisih antara dua set.

PENGAKSESAN SET

1. Tidak Berurutan: Item dalam set tidak memiliki urutan tertentu. Ini berarti bahwa Anda tidak dapat mengakses item dalam set menggunakan indeks.
2. Unik: Set tidak dapat memiliki item yang sama lebih dari satu kali. Setiap item dalam set harus unik.
3. Tidak Berubah (Immutable): Set adalah struktur data yang mutable, yang berarti Anda dapat menambahkan atau menghapus item dari set, tetapi Anda tidak dapat mengubah item yang sudah ada.
4. Cara Akses Item: Karena set tidak berurutan, Anda tidak dapat mengakses item dalam set menggunakan indeks. Sebagai gantinya, Anda dapat menggunakan loop for untuk mengiterasi setiap item dalam set.
5. Menambah dan Menghapus Item: Anda dapat menambahkan item ke dalam set menggunakan metode `add()` dan menghapus item menggunakan metode `remove()` atau `discard()`. Metode `remove()` akan menghasilkan `KeyError` jika item yang ingin dihapus tidak ada dalam set, sedangkan metode `discard()` tidak akan menghasilkan error.
6. Frozenset: Python juga memiliki tipe data frozenset yang mirip dengan set, tetapi tidak dapat diubah setelah dibuat. Anda dapat membuat frozenset dengan menggunakan fungsi `frozenset()`.
7. Metode dan Operasi Set Lainnya: Terdapat berbagai metode lain yang dapat digunakan pada set, seperti `clear()` untuk menghapus semua item dari set, `copy()` untuk membuat salinan set, dan `pop()` untuk menghapus dan mengembalikan item acak dari set. Selain itu, Anda juga dapat menggunakan operator `in` untuk memeriksa keanggotaan item dalam set.

OPERASI PADA SET

Berikut adalah beberapa operasi umum yang dapat Anda lakukan dengan Set dalam Python:

1. Membuat Set:

Set dapat dibuat dengan meletakkan item di dalam tanda kurung kurawal {}. Ini dapat mencakup elemen-elemen dari tipe data apa pun, termasuk campuran tipe data. Misalnya, {1, 2, 3, 'a', 'b', 'c'}.

```
my_set = {1, 2, 3, 4, 5}

mixed_set = {1, 'a', True, 3.14}
```

2. Menambahkan Item:

Pengguna dapat menambahkan item ke dalam Set menggunakan metode add(). Misalnya, jika kita memiliki set my_set = {1, 2, 3} dan kita ingin menambahkan angka 4, kita dapat melakukannya dengan my_set.add(4).

```
my_set.add(6)
print(my_set) # Output: {1, 2, 3, 4, 5, 6}
```

3. Menghapus Item:

Pengguna dapat menghapus item dari Set menggunakan metode discard() atau remove(). Perbedaannya, jika item yang dihapus tidak ada dalam Set, discard() tidak akan menimbulkan error, sementara remove() akan menimbulkan error. Misalnya, jika kita ingin menghapus angka 3 dari my_set, kita bisa menggunakan my_set.discard(3).

```
my_set.discard(3)
print(my_set) # Output: {1, 2, 4, 5, 6}
```

4. Mengecek Keanggotaan:

Pengguna dapat menggunakan operator in untuk memeriksa apakah sebuah item ada dalam Set. Ini berguna untuk menghindari kesalahan saat mencoba mengakses item yang tidak ada dalam Set. Misalnya, if 5 in my_set: akan menghasilkan True jika 5 ada dalam my_set.

```
if 5 in my_set:
    print("5 ada dalam Set")
```

5. Gabungan Set:

Pengguna dapat menggabungkan dua Set menggunakan metode union() atau operator |. Ini akan mengembalikan Set baru yang berisi semua elemen dari kedua Set. Misalnya, union_set = my_set.union(other_set).

```
other_set = {4, 5, 6, 7}
union_set = my_set.union(other_set)
print(union_set) # Output: {1, 2, 4, 5, 6, 7}
```

6. Panjang Set:

Panjang atau jumlah item dalam Set dapat dihitung menggunakan fungsi `len()`. Misalnya, `print("Panjang Set:", len(my_set))` akan mencetak panjang Set `my_set`.

```
print("Panjang Set:", len(my_set))
```

7. Menghapus Semua Item:

Untuk menghapus semua item dari Set, Anda dapat menggunakan metode `clear()`. Misalnya, `my_set.clear()` akan menghapus semua item dari `my_set`.

```
my_set.clear()
print(my_set) # Output: set()
```

8. Iterasi Set:

Pengguna dapat melakukan iterasi atau loop melalui item-item dalam Set menggunakan loop `for`. Misalnya, untuk mencetak semua item dalam `my_set`

```
for item in my_set:
    print(item)
```

Set sangat berguna ketika Anda perlu menyimpan koleksi item yang unik dan tidak memerlukan urutan tertentu. Dengan operasi yang fleksibel dan mudah digunakan, Set adalah struktur data yang penting dalam pemrograman Python.

BAGIAN 2: LATIHAN MANDIRI

SOAL 12.1

- Source Code :

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

apps = {}
for i in range(n):
    kategori = str(input(f"Masukkan nama kategori ke-{i+1}: "))
    set_apps = set()
    for j in range(5):
        app = str(input(f"Masukkan nama aplikasi ke-{j+1}: "))
        set_apps.add(app)
    apps[kategori] = set_apps

list_apps = [apps[kategori] for kategori in apps]
hasil = set.intersection(*list_apps)

print("Aplikasi yang ada di semua kategori:", hasil)
```

- Output :

```
PS C:\Tugas> & C:/Users/ASUS/AppData/Local/Programs/Python/Python312/python.exe "c:/Tugas/Latihan 12,1.py"
Masukkan jumlah kategori: 1
Masukkan nama kategori ke-1: Game
Masukkan nama aplikasi ke-1: Valorant
Masukkan nama aplikasi ke-2: COD
Masukkan nama aplikasi ke-3: POU
Masukkan nama aplikasi ke-4: BOO
Masukkan nama aplikasi ke-5: SUBWAY SURFE
Aplikasi yang ada di semua kategori: {'COD', 'POU', 'Valorant', 'BOO', 'SUBWAY SURFE'}
PS C:\Tugas>
```

- Penjelasan :

Kode ini meminta pengguna memasukkan jumlah kategori, nama kategori, dan nama aplikasi dalam setiap kategori. Kemudian, program membuat kamus `apps` yang berisi set aplikasi untuk setiap kategori. Selanjutnya, kode membuat daftar `list_apps` yang berisi setiap aplikasi untuk setiap kategori, dan kemudian menggunakan `set.intersection(*list_apps)` untuk mencari aplikasi yang ada di semua kategori. Hasilnya dicetak ke layar.

SOAL 12.2

- Source Code :

```
list_data = input("Masukkan data dalam list (pisahkan dengan koma):")
list_data = list(list_data.split(","))
set_data = set(list_data)

print("\nList sebelum konversi:", list_data)
print("Set setelah konversi:", set_data)

set_data = input("\nMasukkan data dalam set (pisahkan dengan koma):")
set_data = set(set_data.split(","))
list_data = list(set_data)

print("\nSet sebelum konversi:", "{" + ", ".join(set_data) + "}")
print("List setelah konversi:", list_data)

tuple_data = tuple(input("\nMasukkan data dalam tuple (pisahkan dengan koma):")
tuple_data = tuple(tuple_data.split(","))
set_data = set(tuple_data)

print("\nTuple sebelum konversi:", tuple_data)
print("Set setelah konversi:", set_data)

set_data = input("\nMasukkan data dalam set (pisahkan dengan koma):")
set_data = set(set_data.split(","))
tuple_data = tuple(set_data)

print("\nSet sebelum konversi:", "{" + ", ".join(set_data) + "}")
print("Tuple setelah konversi:", tuple_data)
```

- Output :

```
PS C:\Tugas> & C:/Users/ASUS/AppData/Local/Programs/Python/Python312/python.exe "c:/Tugas/Latihan 12,2.py"
Masukkan data dalam list (pisahkan dengan koma): 1, 2

List sebelum konversi: ['1', '2']
Set setelah konversi: {'2', '1'}

Masukkan data dalam set (pisahkan dengan koma): 2, 1

Set sebelum konversi: {2, 1}
List setelah konversi: ['2', '1']

Masukkan data dalam tuple (pisahkan dengan koma): 3, 1

Tuple sebelum konversi: ('3', '1')
Set setelah konversi: {'1', '3'}

Masukkan data dalam set (pisahkan dengan koma): 1, 5

Set sebelum konversi: {1, 5}
Tuple setelah konversi: ('1', '5')
```

- Penjelasan :

Pertama, input berupa string yang dipisahkan oleh koma diubah menjadi list menggunakan `split(",")`, kemudian diubah menjadi set. Kedua, input berupa string yang dipisahkan oleh koma diubah menjadi set menggunakan `split(",")`, kemudian diubah menjadi list. Ketiga, input berupa string yang dipisahkan oleh koma diubah menjadi tuple menggunakan `split(",")`, kemudian diubah menjadi set. Keempat, input berupa string yang dipisahkan oleh koma diubah menjadi set menggunakan `split(",")`, kemudian diubah menjadi tuple. Setelah itu, program mencetak hasil konversi.

SOAL 12.3

- Source Code :

```
def baca_konten_berkas(nama_file):
    try:
        with open(nama_file, 'r') as file:
            konten = file.read()
            return konten.lower().split()
    except FileNotFoundError:
        print(f"Berkas '{nama_file}' tidak ditemukan.")
    except PermissionError:
        print(f"Tidak dapat membaca berkas '{nama_file}'.")

def cari_kata_umum(file_pertama, file_kedua):
    kata_dalam_file_pertama = baca_konten_berkas(file_pertama)
    kata_dalam_file_kedua = baca_konten_berkas(file_kedua)

    if kata_dalam_file_pertama and kata_dalam_file_kedua:
        kata_umum = set(kata_dalam_file_pertama) & set(kata_dalam_file_kedua)
        return kata_umum
    return set()

def program_utama():
    file_pertama = input("Masukkan nama berkas pertama: ")
    file_kedua = input("Masukkan nama berkas kedua: ")

    kata_umum = cari_kata_umum(file_pertama, file_kedua)

    if kata_umum:
        print("Kata-kata yang muncul pada kedua berkas:")
        for kata in kata_umum:
            print(kata)

if __name__ == "__main__":
    program_utama()
```

- Output :

```
PS C:\Tugas> & C:/Users/ASUS/AppData/Local/Programs/Python/Python312/python.exe "c:/Tugas/Latihan 12,3.py"
Masukkan nama berkas pertama: lat3.txt
Masukkan nama berkas kedua: lat3-2.txt
Kata-kata yang muncul pada kedua berkas:
a
b
c
e
d
PS C:\Tugas> |
```

- Penjelasan :

Program ini mencari kata-kata umum di dua berkas teks. Fungsi `baca_konten_berkas` membaca berkas dan mengembalikan daftar kata-kata dalam huruf kecil. Fungsi `cari_kata_umum` menggunakan `baca_konten_berkas` untuk kedua berkas dan mengembalikan set kata-kata yang umum. Program utama meminta dua nama berkas teks, mencari kata-kata umum, dan mencetaknya. Program menangani eksepsi `FileNotFoundError` dan `PermissionError`.

- Link GitHub :

<https://github.com/Jamesmarvins/Tugas-Alpro-12.git>