

PRAKTIKUM PEMROGRAMAN KOMPUTER

PERTEMUAN VI

TIPE DATA SET



Disusun oleh:

Nama : Ahmad Fawwas Dzulfahmi
NIM : 25/567941/SV/27316
Kelas : RI1A2
Dosen Pengampu : Ir. Yuris Mulya Saputra, S.T., M.Sc., Ph.D., IPM.

PROGRAM STUDI D-IV TEKNOLOGI REKAYASA INTERNET

DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA

SEKOLAH VOKASI

UNIVERSITAS GADJAH MADA

2025

DAFTAR ISI

DAFTAR ISI.....	1
DAFTAR GAMBAR.....	2
BAB I PENDAHULUAN.....	3
1.1 Latar Belakang.....	3
1.2 Rumusan Masalah	3
1.3 Tujuan.....	3
BAB II PEMBAHASAN.....	4
2.1 Tugas 1	4
2.2 Tugas 2	5
2.3 Tugas 3	6
2.4 Tugas 4	7
BAB III PENUTUP.....	8
3.1 Kesimpulan.....	8
DAFTAR PUSTAKA.....	9

DAFTAR GAMBAR

Tugas 1 : Source Code	4
Tugas 1 : Output.....	4
Tugas 2 : Source Code	5
Tugas 2 : tambahan source code	5
Tugas 2 : output dari kode yang sudah ditambahkan	5
Tugas 3 : Source Code	6
Tugas 3 : Source Code dengan modifikasi	6
Tugas 3 : Output.....	6
Tugas 4 : Source code angka ada	7
Tugas 4 : output dari source code pertama	7
Tugas 4 : source code kedua	7
Tugas 4 : output code	7

BAB I

PENDAHULUAN

1.1 Latar Belakang

Set merupakan salah satu tipe data yang dapat digunakan untuk mengelola kumpulan data. Berbeda dengan list atau tuple, set memiliki keunikan dimana ia secara otomatis menghilangkan elemen duplikat dan juga sifatnya yang tidak terurut, sehingga posisi elemen bisa berubah-ubah setiap kali diakses. Set juga memiliki sifat mutable kita bisa menambahkan atau menghapus elemen, tetapi tidak bisa mengubah set itu sendiri. Selain itu set sendiri menggunakan *hashing* yang dapat membuat set lebih efisien saat melakukan pencarian, memasukan data maupun menghapus data.

Python juga menyediakan berbagai operasi himpunan yang mirip dengan matematika, seperti union dengan operator ($|$), intersection dengan ($&$), selisih atau difference dengan ($-$), dan selisih simetris dengan ($^$). Untuk memanipulasi data, set memiliki metode praktis seperti `add()` untuk menambah elemen, `remove()` untuk menghapus dengan spesifik, `discard()` untuk menghapus tanpa error, serta `pop()` untuk mengambil elemen acak.

1.2 Rumusan Masalah

Adapun Rumusan Masalah Sebagai Berikut:

1. Melakukan analisa terhadap sebuah variabel *set*.
2. Membuat program untuk menambahkan dan menghapus pada sebuah himpunan *set*.
3. Membuat sebuah program singkat untuk mengelola sebuah tipe data *set*.
4. Memnjalankan perintah hapus dari dua syntax yang berbeda.

1.3 Tujuan

Adapun Tujuan dari Laporan Praktikum sebagai berikut:

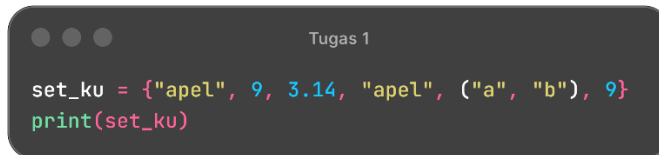
1. Mahasiswa mampu menjelaskan konsep tipe data set pada python.
2. Mahasiswa mampu menjelaskan cara menggunakan tipe data set pada python.
3. Mahasiswa dapat menerapkan konsep yang berkaitan dengan tipe data set pada python.

BAB II

PEMBAHASAN

2.1 Tugas 1

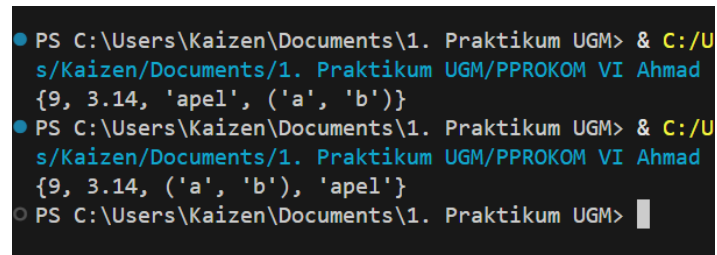
Analisis hasil kode di bawah dan jelaskan berdasarkan karakteristik tipe data set



```
set_ku = {"apel", 9, 3.14, "apel", ("a", "b"), 9}
print(set_ku)
```

Tugas 1 : Source Code

Jawaban :



```
PS C:\Users\Kaizen\Documents\1. Praktikum UGM> & C:/U
s/Kaizen/Documents/1. Praktikum UGM/PPROKOM VI Ahmad
{9, 3.14, 'apel', ('a', 'b')}
PS C:\Users\Kaizen\Documents\1. Praktikum UGM> & C:/U
s/Kaizen/Documents/1. Praktikum UGM/PPROKOM VI Ahmad
{9, 3.14, ('a', 'b'), 'apel'}
PS C:\Users\Kaizen\Documents\1. Praktikum UGM> █
```

Tugas 1 : Output

Kode di atas adalah sebuah perintah yang digunakan untuk menampilkan hasil *print* dari sebuah data tipe set. Karena sifat dari *set* sendiri yang tidak dapat memiliki elemen yang sama dan index yang tidak berurut, membuat hasil *print* dapat berubah rubah setiap program dijalankan seperti yang terlihat pada gambar

2.2 Tugas 2

Diketahui suatu set sebagai berikut :

```
nilai = set({3,6,9,2,5,7})
```

Tugas 2 : Source Code

Buatlah program untuk menambahkan dan menghapus elemen baru pada himpunan 'nilai' di atas sehingga himpunan setelah ditambahkan menjadi :

- nilai = set ({1,2,3,4,5,6,7,8,9,10}) à setelah proses penambahan
- nilai = set({6,9,2}) à setelah proses penghapusan

```
nilai1 = set({3,6,9,2,5,7})

nilai1.update({1,2,4,8,10})
penghapusan = nilai1.intersection({6, 9, 2})

print(nilai1)
print(penghapusan)
```

Tugas 2 : tambahan source code

```
PS C:\Users\Kaizen\Documents\1. Praktikum UGM\PPR
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
{9, 2, 6}
PS C:\Users\Kaizen\Documents\1. Praktikum
```

Tugas 2 : output dari kode yang sudah ditambahkan

Penjelasan :

Untuk menambahkan data atau memasukan elemen kedalam set kita dapat menggunakan .update() maupun add(), karena kita akan memasukan beberapa elemen maka kita menggunakan update(), add () sendiri digunakan jika kita hanya ingin memasukan 1 elemen saja. Setelah itu untuk menghilangkan objek lainnya dan menampilkan hanya angka 6 9 2 kita bisa menggunakan intersection. Karena tipe data yang terdapat pada set sama yaitu integer maka sistem akan secara otomatis mengurutkannya.

2.3 Tugas 3

Diberikan dua set:

```
set_A = {20, 30, 40, 50, 60}
set_B = {25, 30, 35, 40, 45}
set_C = {30, 40, 50, 70, 80}
set_D = {40, 50, 60, 90, 100}
```

Tugas 3 : Source Code

- Irisan set_A, set_C, dan set_D
- Gabungan set_A dan set_B, lalu selisihkan dengan set_D
- Selisih simetris dari set_B dan set_C
- Gabungkan set_A dengan set_B serta set_C dengan set_D, lalu buat irisan dari kedua gabungan itu.

```
Tugas 3

set_A = {20, 30, 40, 50, 60}
set_B = {25, 30, 35, 40, 45}
set_C = {30, 40, 50, 70, 80}
set_D = {40, 50, 60, 90, 100}

irisan = set_A & set_B & set_C
gabungan1 = set_A | set_B
gabungan2 = set_C | set_D
selisih = gabungan1.difference(set_D)
selisih_simetris = set_B ^ set_C
selisih_gabungan = gabungan1.difference(gabungan2)

print(irisan)
print(selisih)
print(selisih_simetris)
print(selisih_gabungan)
```

Tugas 3 : Source Code dengan modifikasi

```
PS C:\Users\Kaizen\Documents\1. Praktikum UGM> &
s/Kaizen/Documents/1. Praktikum UGM/PPROKOM VI A
{40, 30}
{35, 45, 20, 25, 30}
{80, 50, 35, 70, 25, 45}
{25, 35, 20, 45}
PS C:\Users\Kaizen\Documents\1. Praktikum UGM> [
```

Tugas 3 : Output

Penjelasan :

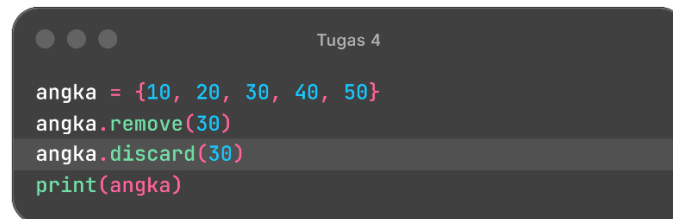
Untuk membuat irisan kita dapat menggunakan simbol intersection (&), untuk menggabungkan sebuah set kita dapat menggunakan simbol union (|). Untuk menghitung selisih simetris kita dapat menggunakan simbol (^). Lalu yang terakhir kita gabungkan a dengan b lalu c dengan d, lalu kita buat irisannya menggunakan perintah difference,

2.4 Tugas 4

Sebuah set memiliki data sebagai berikut: `angka = {10, 20, 30, 40, 50}`. Jelaskan perbedaan antara `remove()` dan `discard()`. Tuliskan dua contoh kode:

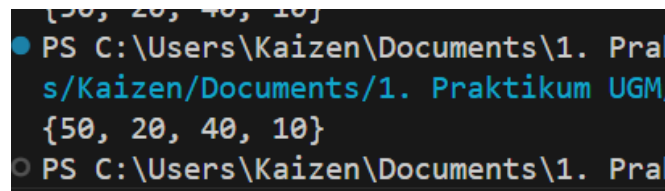
- Contoh menghapus dengan `remove()` dan `discard()` angka 30 dari set.
- Contoh menghapus dengan `remove()` dan `discard()` angka 99 dari set.

Jawaban :



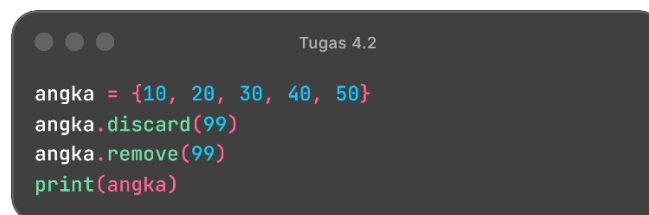
```
angka = {10, 20, 30, 40, 50}
angka.remove(30)
angka.discard(30)
print(angka)
```

Tugas 4 : Source code angka ada



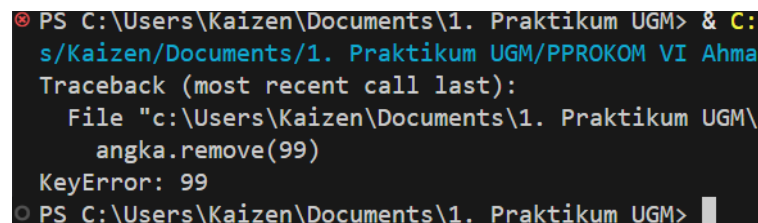
```
[50, 20, 40, 10]
PS C:\Users\Kaizen\Documents\1. Praktikum UGM>
{50, 20, 40, 10}
PS C:\Users\Kaizen\Documents\1. Praktikum UGM>
```

Tugas 4 : output dari source code pertama



```
angka = {10, 20, 30, 40, 50}
angka.discard(99)
angka.remove(99)
print(angka)
```

Tugas 4 : source code kedua



```
PS C:\Users\Kaizen\Documents\1. Praktikum UGM> & C:\Python39\python.exe C:\Users\Kaizen\Documents\1. Praktikum UGM\PPROKOM VI Ahma
Traceback (most recent call last):
  File "c:\Users\Kaizen\Documents\1. Praktikum UGM\PPROKOM VI Ahma.py", line 5, in <module>
    angka.remove(99)
KeyError: 99
PS C:\Users\Kaizen\Documents\1. Praktikum UGM>
```

Tugas 4 : output code

Penjelasan :

Pada bagan pertama hasilnya memperlihatkan kalau angka 30 berhasil kita hilangkan lalu kita bisa coba menggantinya dengan angka yang lain, disini karena perintahnya menggunakan angka 99 maka bisa kita masukkan dan nanti akan menghasilkan `KeyError: 99` dikarenakan sifat `remove` yang menghapus secara spesifik jadi jika variabel `ang` ingin dihapus tidak ada akan menghasilkan `KeyError`

BAB III

PENUTUP

3.1 Kesimpulan

Tipe data set pada python memiliki karakteristik unik yang membedakannya dari tipe data lainnya. Karena set akan menghapus secara otomatis elemen yang sama atau ter duplikasi serta tidak menjaga urutan elemen, sehingga cocok digunakan untuk menganalisa data yang tergolong unik atau tidak jelas urutannya. Selain itu python juga dilengkapi dengan operasi himpunan yang dapat digunakan untuk mempermudah mengelola tipe data set. Seperti union, intersection, difference, dan symmetric difference.

Selain itu terdapat dua perbedaan yang mendasar diantara metode `remove()` dan `discard()` yang dapat terlihat dengan jelas. ketika kita menggunakan `remove()` akan menghasilkan `KeyError` sat elemen tidak ada, sedangkan `discard()` lebih aman karena walaupun elemen yang ingin dihapus tidak ada tidak memunculkan error. Kita juga bisa menambah elemen menggunakan `update()` berbanding terbalik dengan `add()`, `update()` dapat menambah beberapa elemen sekaligus sedangkan untuk `add()` hanya dapat menambah satu elemen saja.

DAFTAR PUSTAKA

- GeeksforGeeks. (2016, May 16). *Sets in Python*. GeeksforGeeks.
<https://www.geeksforgeeks.org/python/sets-in-python/>
- Juliano Luiz Faccioni. (2023, September 4). *An In-Depth Guide to Working with Python Sets*. LearnPython.com. <https://learnpython.com/blog/python-sets/>
- python. (n.d.-a). 5. *Data Structures*. Python Documentation. Retrieved September 21, 2025, from <https://docs.python.org/3/tutorial/datastructures.html#sets>
- python. (n.d.-b). *Built-in Types — Python 3.9.1 documentation*. Docs.python.org. Retrieved September 21, 2025, from <https://docs.python.org/3/library/stdtypes.html#set-types-set-frozenset>
- W3Schools. (2020). *Python Sets*. W3schools.com.
https://www.w3schools.com/python/python_sets.asp