

LAPORAN PRAKTIKUM  
PEMOGRAMAN KOMPUTER  
TIPE DATA SET



Disusun Oleh:

Nama : Noni Cindy Klaudia Matatar  
NIM : 24/545671/SV/25729  
Kelas : RI1B1  
Dosen Pengampu : Yuris Mulya Saputra, S.T.,M.Sc.,Ph.

PROGRAM STUDI D-IV TEKNOLOGI REKAYASA INTERNET  
DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA  
SEKOLAH VOKASI  
UNIVERSITAS GADJAH MADA  
YOGYAKARTA

2024

## DAFTAR ISI

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

## DAFTAR GAMBAR

Gambar 2. 1 Tugas 1 .....	5
Gambar 2. 2 Tugas 2 .....	6
Gambar 2. 3 Tugas 3 .....	7
Gambar 2. 4 Tugas 4 .....	8
Gambar 2. 5 Tugas 5 .....	9
Gambar 2. 6 Tugas 6 .....	10

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Tipe data set adalah struktur data yang digunakan untuk menyimpan koleksi data yang terorganisir. Dalam konteks pemrograman dan analisis data, tipe data set memungkinkan pengelolaan informasi secara efisien dan terstruktur. Setiap elemen dalam data set dapat berupa berbagai tipe data, seperti angka, string, atau objek, dan bisa diakses secara indeks.

Data set sering digunakan dalam analisis statistik, pembelajaran mesin, dan basis data. Mereka memungkinkan pengelolaan data dalam jumlah besar dengan cara yang memudahkan pemrograman untuk melakukan operasi seperti pencarian, pengurutan, dan pengelompokan.

Tipe data set juga mencakup fitur seperti kemampuan untuk menangani data yang hilang, pengelompokan data, dan penggabungan berbagai sumber data. Dengan berkembangnya teknologi dan kebutuhan akan data yang lebih kompleks, tipe data set semakin penting dalam pengembangan aplikasi dan penelitian.

### **1.2 Rumusan Masalah**

1. Bagaimana berbagai tipe data dalam set dapat mempengaruhi kapasitas penyimpanan dan kecepatan akses data dalam aplikasi basis data?
2. Apa dampak dari pemilihan tipe data yang tidak tepat terhadap kualitas dan keakuratan analisis data dalam proyek data science?
3. Sejauh mana penggunaan tipe data yang berbeda dapat mempengaruhi performa pemrosesan dan efisiensi algoritma dalam pemrograman dan pengembangan perangkat lunak?

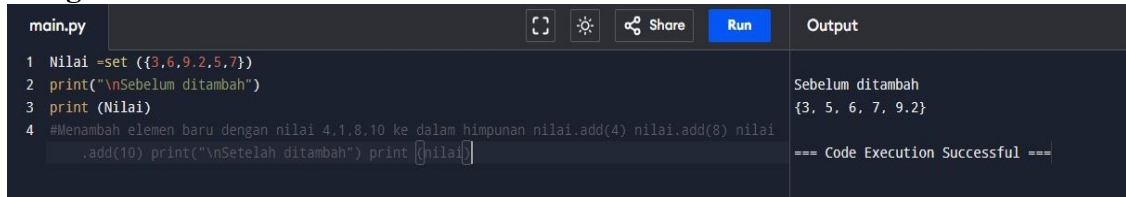
### **1.3 Tujuan**

1. Untuk memahami bagaimana tipe data yang berbeda dalam set mempengaruhi efisiensi penyimpanan dan kecepatan akses dalam aplikasi basis data
2. Untuk memahami bagaimana pilihan tipe data yang tidak tepat dapat memengaruhi kualitas dan keakuratan analisis data dalam proyek data science.
3. Untuk mengeksplorasi bagaimana pemilihan tipe data yang tepat dapat mempengaruhi performa dan efisiensi algoritma dalam pemrograman.

## BAB II

### PEMBAHASAN

#### 2.1 Tugas 1



The screenshot shows a code editor with a file named 'main.py'. The code defines a set 'Nilai' with elements {3, 6, 9, 2, 5, 7}, prints a message, and then adds elements 4, 1, 8, and 10 to the set. The output shows the set before and after the additions, and a success message.

```
main.py
1 Nilai = set ({3,6,9,2,5,7})
2 print("\nSebelum ditambah")
3 print (Nilai)
4 #Menambah elemen baru dengan nilai 4,1,8,10 ke dalam himpunan nilai.add(4) nilai.add(8) nilai
  .add(10) print("\nSetelah ditambah") print (Nilai)
```

Output

```
Sebelum ditambah
{3, 5, 6, 7, 9, 2}

=== Code Execution Successful ===
```

*Gambar 2. 1 Tugas 1*

Analisis:

1. Baris pertama:

- Membuat himpunan (set) Bernama Nilai yang berisi elemen-elemen berupa angka: 3,6,9,2,5, dan 7. Himpunan adalah tipe data yang tidak memperbolehkan elemen duplikat dan tidak memiliki urutan tetap.

2. Baris kedua:

- Menampilkan teks “Senelum ditambah” sebagai penanda untuk menampilkan isi himpunan sebelum elemen baru ditambahkan

3. Baris ketiga:


- Menampilkan isi dari himpunan Nilai.

Hasilnya adalah {3, 5, 6, 7, 9, 2.}, dengan elemen disusun tanpa urutan tertentu karena himpunan tidak teratur.

4. Output yang ditampilkan:

- Output menampilkan “Sebelum ditambah” dan himpunan {3, 5, 6, 7, 9, 2.}, yang mencerminkan urutan acak karena bersifat himpunan.

## 2.2 Tugas 2



```
main.py
1 nilai = set({3,6,9,2,5,7})
2 print("\nSebelum dihapus")
3 print (nilai)
4 #Menambah elemen baru dengan nilai 4,1,8,10 ke dalam himpunan nilai.add(4) nilai.add(8) nilai
  .add(10)
5 #Menghapus 3,5,7 dari himpunan nilai.remove(3) nilai.remove(5) nilai.remove(7)
6 print("\nSetelah ditambah")
7 print (nilai)
8 print("\nSetelah dihapus")
9 print (nilai)
```

Output

Sebelum dihapus  
{2, 3, 5, 6, 7, 9}

Setelah ditambah  
{2, 3, 5, 6, 7, 9}

Setelah dihapus  
{2, 3, 5, 6, 7, 9}

=== Code Execution Successful ===

Gambar 2. 2 Tugas 2

Analisis:

1. Baris 1:

Ini mendefinisikan sebuah himpunan (set) dengan elemen {3, 6, 9, 2, 7}. Himpunan tidak memiliki elemen duplikat, dan urutannya tidak terjamin

2. Baris 2-3:

Baris ini mencetak kalimat “Sebelum dihapus” dan mencetak nilai dari himpunan nilai. Hasil yang dicetak adalah: {2, 3, 5, 6, 7, 9}. Ini menunjukkan bahwa himpunan mengurutkan elemen secara otomatis

3. Baris 4 (Komentar):

Komentar ini menyebutkan rencana penambahan elemen 4, 1, 8, dan 10 ke dalam himpunan, tetapi baris ini tidak dieksekusi karena hanya komentar.

4. Baris 5 (Komentar):

Ini juga merupakan komentar yang menyebutkan penghapusan 3, 5, dan 7, namun baris ini tidak dieksekusi.

5. Baris 6-7:

Ini mencetak “Setelah ditambah” dan mencetak Kembali himpunan nilai. Hasil yang di cetak adalah : {2, 3, 5, 6, 7, 9}. Tidak ada perubahan pada himpunan, karena elemen tidak benar-benar ditambahkan (baris untuk penambahan hanya untuk komentar).

6. Baris 8-9 :

Ini mencetak “Setelah dihapus” dan mencetak himpunan nilai lagi. Hasilnya tetap sama:

{2, 3, 5, 6, 7, 9}. Tidak ada elemen yang dihapus karena baris untuk penghapusan juga hanya komentar.

## 2.3 Tugas 3

```
main.py [ ] [ ] [ ] Share Run Output
1 A = {100, 7, 8}
2 B = {200, 4, 5}
3 C = {300, 2, 3}
4 D = {100, 200, 300}
5 print(A.intersection(D))
6 print(B.intersection(D))
7 print(C.intersection(D))
8 print(A.intersection(B, C, D))
```

```
{100}
{200}
{300}
set()

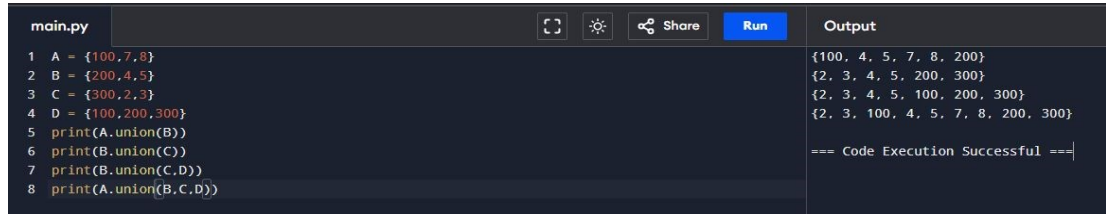
=== Code Execution Successful ===
```

Gambar 2. 3 Tugas 3

Analisis:

1. Ada empat himpunan yang didefinisikan:
  - $A = \{100, 7, 8\}$
  - $B = \{200, 4, 5\}$
  - $C = \{300, 2, 3\}$
  - $D = \{100, 200, 300\}$
2. Pada garis 5 hingga 8, program menghitung irisan (intersection) antara himpunan-himpunan tersebut dengan D.
  - $A.intersection(D)$  menghasilkan  $\{100\}$ , karena 100 adalah satu-satunya elemen yang ada di A dan D.
  - $B.intersection(D)$  menghasilkan  $\{200\}$ , karena 200 adalah satu-satunya elemen yang ada di B dan D.
  - $C.intersection(D)$  menghasilkan  $\{300\}$ , karena 300 adalah satu-satunya elemen yang ada di C dan D.
  - $A.intersection(B, C, D)$  menghasilkan  $set()$ , karena tidak ada elemen yang sama di keempat himpunan.
  - Output program tersebut menampilkan hasil dari fungsi `intersection()` yang sesuai dengan analisis di atas.

## 2.4 Tugas 4



```
main.py
1 A = {100,7,8}
2 B = {200,4,5}
3 C = {300,2,3}
4 D = {100,200,300}
5 print(A.union(B))
6 print(B.union(C))
7 print(B.union(C,D))
8 print(A.union(B,C,D))
```

Output

```
{100, 4, 5, 7, 8, 200}
{2, 3, 4, 5, 200, 300}
{2, 3, 4, 5, 100, 200, 300}
{2, 3, 100, 4, 5, 7, 8, 200, 300}

=== Code Execution Successful ===
```

Gambar 2. 4 Tugas 4

Analisis:

1. Himpunan yang didefinisikan:

$$A = \{100, 7, 8\}$$

$$B = \{200, 4, 5\}$$

$$C = \{300, 2, 3\}$$

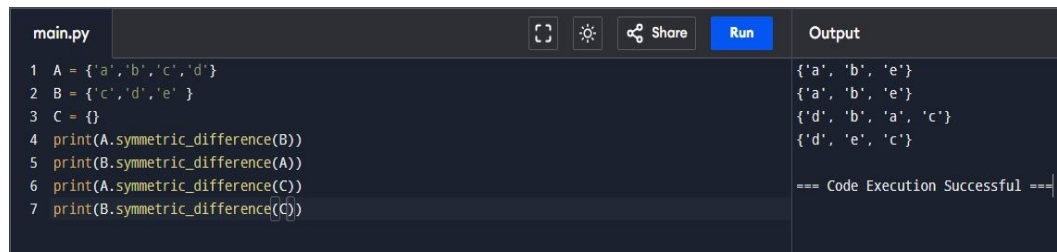
$$D = \{100, 200, 300\}$$

2. Pada baris 5 hingga 8, program menghitung gabungan (union) dari beberapa himpunan.

- Baris 5: `A.union(B)` menghasilkan `{100, 4, 7, 8, 200}`, yang merupakan gabungan elemen dari himpunan A dan B.
- Baris 6: `B.union(C)` menghasilkan `{2, 3, 4, 5, 200, 300}`, yang merupakan gabungan elemen dari himpunan B dan C.
- Baris 7: `B.union(C, D)` menghasilkan `{2, 3, 4, 5, 100, 200, 300}`, yang merupakan gabungan elemen dari himpunan B, C dan D.
- Baris 8: `A.union(B, C, D)`  
Menghasilkan `{2, 3, 100, 4, 5, 7, 8, 200, 300}`, yang merupakan gabungan elemen dari himpunan A, B, C, dan D.
- Output yang ditampilkan:  
Output menampilkan hasil gabungan dari masing-masing operasi `union()` yang sesuai dengan analisis diatas.



## 2.5 Tugas 5



The screenshot shows a Jupyter Notebook with a file named 'main.py'. The code defines three sets: A = {'a', 'b', 'c', 'd'}, B = {'c', 'd', 'e'}, and C = {}. It then prints the symmetric difference of A and B, B and A, A and C, and B and C. The output shows that A.symmetric\_difference(B) and B.symmetric\_difference(A) both return {'a', 'b', 'e'}, A.symmetric\_difference(C) returns {'a', 'b', 'c', 'd'}, and B.symmetric\_difference(C) returns {'c', 'd', 'e'}. The notebook also shows a 'Run' button and a 'Share' icon.

```
main.py
1 A = {'a', 'b', 'c', 'd'}
2 B = {'c', 'd', 'e'}
3 C = {}
4 print(A.symmetric_difference(B))
5 print(B.symmetric_difference(A))
6 print(A.symmetric_difference(C))
7 print(B.symmetric_difference(C))
```

Output

```
{'a', 'b', 'e'}
{'a', 'b', 'e'}
{'d', 'b', 'a', 'c'}
{'d', 'e', 'c'}

=== Code Execution Successful ===
```

Gambar 2. 5 Tugas 5

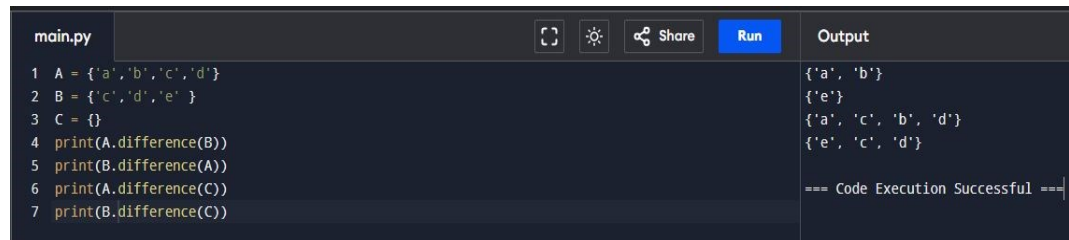
Analisis:

1. Himpunan A berisi elemen { 'a', 'b', 'c', 'd' }
2. Himpunan B berisi elemen { 'c', 'd', 'e' }
3. Himpunan C adalah himpunan kosong { }

Kemudian, kode mencetak hasil perbedaan simetris antara himpunan-himpunan tersebut:

1. A.symmetric\_difference(B) menghasilkan elemen yang ada di A atau B, tetapi tidak di kedua himpunan: { 'a', 'b', 'e' }
2. B.symmetric\_difference(A) juga mengembalikan { 'a', 'b', 'e' } karena perbedaan simetris bersifat komutatif.
3. A.symmetric\_difference(C) mengembalikan elemen yang ada di A, karena C adalah himpunan kosong: { 'a', 'b', 'c', 'd' }
4. B.symmetric\_difference(C) mengembalikan elemen yang hanya ada di B, yaitu: { 'c', 'd', 'e' }

## 2.6 Tugas 6



The screenshot shows a Python IDE with a file named 'main.py'. The code defines three sets: A = {'a', 'b', 'c', 'd'}, B = {'c', 'd', 'e'}, and C = {}. It then prints the results of A.difference(B), B.difference(A), A.difference(C), and B.difference(C). The output on the right shows the results: {'a', 'b'}, {'e'}, {'a', 'c', 'b', 'd'}, and {'e', 'c', 'd'}. The code execution is successful.

```
main.py
1 A = {'a', 'b', 'c', 'd'}
2 B = {'c', 'd', 'e'}
3 C = {}
4 print(A.difference(B))
5 print(B.difference(A))
6 print(A.difference(C))
7 print(B.difference(C))

Output
{'a', 'b'}
{'e'}
{'a', 'c', 'b', 'd'}
{'e', 'c', 'd'}
=== Code Execution Successful ===
```

Gambar 2. 6 Tugas 6

Analisis:

1. Baris 4: A.difference(B)  
Set A: {'a', 'b', 'c', 'd'}  
Set B: {'c', 'd', 'e'}  
Elemen yang hanya ada di A dan tidak ada di B adalah {'a', 'b'}.
  2. Baris 5: B.difference(A)  
Set B: {'c', 'd', 'e'}  
Set A: {'a', 'b', 'c', 'd'}  
Elemen yang hanya ada di B dan tidak ada di A adalah {'e'}.
  3. Baris 6: A.difference(C)  
Set A: {'a', 'b', 'c', 'd'}  
Set C: {} (kosong)  
Karena C kosong, seluruh elemen di A ditampilkan: {'a', 'c', 'b', 'd'}.
  4. Baris 7: B.difference(C)  
Set B: {'c', 'd', 'e'}  
Set C: {} (kosong)  
Karena C kosong, seluruh elemen di B ditampilkan: {'e', 'c', 'd'}.
- Outputnya sesuai dengan perbandingan elemen antara set yang berbeda, dan program dijalankan dengan sukses.

## **BAB III**

### **PENUTUP**

#### **3.1 Kesimpulan**

Tipe data set dalam pemograman (terutama di python) adalah Kumpulan data yang tidak berurutan dan tidak mengizinkan duplikasi. Set mirip dengan himpunan dalam matematika, di mana setiap elemen hanya muncul satu kali. Elemen-elemen dalam set bisa berupa berbagai tipe data, teteapi set itu sendiri bersifat tidak berurutan, sehingga tidak bisa diakses menggunakan indeks. Operasi umum pada set meliputi penambahan, penghapusan elemen, serta operasi himpunan seperti union, intersection, dan difference.

## DAFTAR PUSTAKA

Huda Nurul. (2021, Januari 28). *Python Dasar: Tipe Data Set*  
[.https://jagongoding.com/python/dasar/set/](https://jagongoding.com/python/dasar/set/).

Team By. (2024 Maret 1). *Data Set Adalah: Pengertian Jenis dan Contohnya*.  
<https://codingstudio.id/blog/data-set-adalah/#>.