

## Panduan Praktikum Pertemuan ke-2

Dalam Python, tipe data koleksi seperti list, tuple, set, dan dictionary digunakan untuk menyimpan kumpulan data. Masing-masing memiliki karakteristik unik yang membedakan mereka satu sama lain serta penggunaan spesifik berdasarkan kebutuhan. Berikut adalah perbedaan dan persamaan dari tipe data koleksi tersebut dalam konteks koleksi kosong:

### List

- **Empty Collection:** `[]`
- **Deskripsi:** List adalah koleksi yang terurut dan dapat diubah. List memungkinkan duplikasi anggota.
- **Kegunaan:** Cocok untuk kumpulan data yang urutannya penting dan dapat berubah, serta memerlukan kemampuan untuk menyimpan elemen duplikat.
- `my_list = [1, 2, 2, 3]` # List dengan elemen duplikat

### Tuple

- **Empty Collection:** `()`
- **Deskripsi:** Tuple adalah koleksi yang terurut dan tidak dapat diubah. Tuple memungkinkan duplikasi anggota.
- **Kegunaan:** Cocok untuk kumpulan data yang tidak boleh diubah setelah dibuat, seperti konstanta atau menyimpan koleksi elemen dari berbagai tipe data.
- `my_tuple = (1, 2, 3)` # Tuple

### Set

- **Empty Collection:** `set()` atau `{}`
- **Deskripsi:** Set adalah koleksi yang tidak terurut, tidak terindeks, dapat diubah, dan tidak memungkinkan duplikasi anggota.
- **Kegunaan:** Cocok untuk operasi matematika set seperti union, intersection, dan set difference, serta untuk memastikan elemen unik dalam kumpulan.
- `my_set = {1, 2, 3}` # Set tanpa duplikat

### Dictionary

- **Empty Collection:** `{}`
- **Deskripsi:** Dictionary adalah koleksi yang tidak terurut, dapat diubah, dan terindeks. Dictionary menyimpan pasangan kunci:nilai dan tidak memungkinkan kunci duplikat.
- **Kegunaan:** Cocok untuk menyimpan data dalam pasangan kunci:nilai untuk pencarian, pembaruan, dan penghapusan yang cepat berdasarkan kunci.
- `my_dict = {}` # Dictionary kosong

## Panduan Praktikum Pertemuan ke-2

### Persamaan

- **Penyimpanan Koleksi:** Semua tipe data ini digunakan untuk menyimpan kumpulan data dalam berbagai cara.
- **Dapat Diiterasi:** Semua tipe data ini dapat diiterasi, yang berarti Anda dapat menggunakan loop (seperti for loop) untuk mengakses elemen-elemen dalam koleksi.
- **Dinamis:** Kecuali tuple, semua tipe data koleksi lainnya (list, set, dan dictionary) adalah dinamis, artinya ukuran mereka dapat berubah selama eksekusi program.

### Perbedaan

- **Mutabilitas:** List dan dictionary adalah mutable, memungkinkan modifikasi setelah pembuatan. Tuple immutable, tidak dapat diubah setelah dibuat. Set dapat dimodifikasi tetapi tidak mendukung akses berdasarkan indeks atau kunci.
- **Pengindeksan dan Pengurutan:** List dan tuple adalah terurut dan mendukung akses berdasarkan indeks. Dictionary dan set tidak terurut; dictionary diakses berdasarkan kunci, sedangkan set tidak mendukung akses berdasarkan indeks atau kunci.
- **Duplikasi:** List dan tuple memungkinkan duplikasi elemen. Set secara otomatis menghilangkan duplikasi elemen. Dictionary tidak memungkinkan duplikasi kunci tetapi memungkinkan duplikasi nilai.
- **Sintaks untuk Koleksi Kosong:** List dan dictionary menggunakan kurung siku dan kurung kurawal untuk mewakili koleksi kosong (`[]` dan `{}`), sedangkan tuple menggunakan kurung biasa `()` dan set menggunakan `set()`.

### Kesimpulan

- **List** dan **Tuple** mirip karena keduanya bersifat ordered dan dapat berisi duplikat, tetapi list mutable sedangkan tuple immutable.
- **Set** unik karena unordered dan tidak dapat berisi duplikat, berguna untuk operasi matematika seperti union, intersection, dan lainnya.
- **Dictionary** menyimpan data dalam pasangan kunci-nilai, menjadikannya pilihan yang tepat untuk penyimpanan data yang terstruktur dan cepat diakses melalui kunci

**Catatan :** Pemilihan antara tipe data koleksi ini tergantung pada kebutuhan spesifik program, termasuk apakah elemen-elemen perlu diurutkan, apakah modifikasi setelah pembuatan diperlukan, dan apakah duplikasi elemen diperbolehkan.

## Panduan Praktikum Pertemuan ke-2

### Contoh Penggunaan List :

# Mendefinisikan list

```
my_list = [1, 2, 3, 4, 5]
```

```
print(my_list)
```

# List dengan berbagai tipe data

```
mixed_list = [1, "Hello", 3.14, [1, 2, 3], True]
```

```
print(mixed_list)
```

# Menambahkan elemen ke list

```
my_list.append(6) # Menambahkan 6 ke akhir list
```

```
print(my_list)
```

# Mengakses elemen list

```
print(my_list[0]) # Output: 1, mengakses elemen pertama
```

```
print(mixed_list[3][1]) # Output: 2, mengakses elemen kedua dari sublist
```

# Memotong list

```
print(my_list[1:4]) # Output: [2, 3, 4], mengambil elemen kedua hingga keempat
```

# Mengubah elemen list

```
my_list[0] = 10 # Mengubah elemen pertama menjadi 10
```

```
print(my_list)
```

# Menghapus elemen list

```
del my_list[2] # Menghapus elemen ketiga
```

```
print(my_list)
```

# Menggabungkan list

```
another_list = [7, 8, 9]
```

```
combined_list = my_list + another_list # Menggabungkan dua list
```

```
print(combined_list)
```

# Iterasi melalui list

```
for item in my_list:
```

```
    print(item)
```

## Panduan Praktikum Pertemuan ke-2

### Contoh Penggunaan Set :

```
# Mendefinisikan set dengan beberapa elemen
my_set = {1, 2, 3, 4, 5}
print(my_set)

# Mendefinisikan set menggunakan fungsi set()
my_set2 = set([1, 2, 2, 3, 4]) # Menggunakan list sebagai argumen
print(my_set2) # Output: {1, 2, 3, 4} - duplikat dihilangkan

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

# Menghapus elemen dari set
my_set.remove(6) # Jika elemen tidak ada, akan muncul KeyError
print(my_set)

# Menggunakan operasi set
a = {1, 2, 3, 4}
b = {3, 4, 5, 6}

# Union
print(a | b) # Output: {1, 2, 3, 4, 5, 6}

# Intersection
print(a & b) # Output: {3, 4}

# Difference
print(a - b) # Output: {1, 2}

# Symmetric Difference
print(a ^ b) # Output: {1, 2, 5, 6}
```

## Panduan Praktikum Pertemuan ke-2

### Contoh Penggunaan Tuple :

```
# Mendefinisikan tuple
my_tuple = (1, "Hello", 3.14)
print(my_tuple)

# Tuple dengan satu elemen harus diikuti koma
single_element_tuple = (1,)
print(single_element_tuple)

# Tuple tanpa tanda kurung
my_tuple_without_parentheses = 1, 2, 3
print(my_tuple_without_parentheses)

# Mengakses elemen tuple
print(my_tuple[1]) # Output: Hello

# Tuple unpacking
a, b, c = my_tuple
print(a, b, c)

# Tuple tidak dapat diubah (immutable)
# Berikut ini akan menyebabkan error jika dijalankan
# my_tuple[1] = "World" # TypeError: 'tuple' object does not support item assignment

# Menggunakan tuple dalam fungsi yang mengembalikan lebih dari satu nilai
def min_max(items):
    return min(items), max(items)

print(min_max([1, 2, 3, 4, 5])) # Output: (1, 5)
```

## Panduan Praktikum Pertemuan ke-2

### Contoh Penggunaan Dictionary :

# Mendefinisikan dictionary

```
my_dict = {"nama": "John", "usia": 30, "kota": "New York"}  
print(my_dict)
```

# Mengakses nilai menggunakan kunci

```
print(my_dict["nama"]) # Output: John
```

# Menambahkan atau memperbarui pasangan kunci-nilai

```
my_dict["email"] = "john@example.com" # Menambahkan kunci baru 'email'
```

```
my_dict["usia"] = 31 # Memperbarui nilai dari kunci 'usia'
```

```
print(my_dict)
```

# Menghapus pasangan kunci-nilai dengan del atau pop()

```
del my_dict["kota"] # Menghapus kunci 'kota'
```

```
print(my_dict)
```

```
email = my_dict.pop("email") # Menghapus dan mengembalikan nilai dari kunci 'email'
```

```
print(email)
```

```
print(my_dict)
```

# Menggunakan keys(), values(), dan items()

```
print(my_dict.keys()) # Output: dict_keys(['nama', 'usia'])
```

```
print(my_dict.values()) # Output: dict_values(['John', 31])
```

```
print(my_dict.items()) # Output: dict_items([('nama', 'John'), ('usia', 31)])
```

# Iterasi melalui dictionary

```
for key, value in my_dict.items():
```

```
    print(f"Kunci: {key}, Nilai: {value}")
```

## Panduan Praktikum Pertemuan ke-2

### KEGIATAN PRAKTIKUM

Mengerjakan seluruh soal dibawah ini dan dikumpulkan sesuai arahan pada saat praktikum.

#### SOAL LIST :

##### Soal 1: Pembuatan dan Akses List

Buatlah sebuah list yang berisi lima nama buah. Kemudian, akses dan cetak nama buah kedua dan keempat dari list tersebut.

##### Soal 2: Modifikasi List

Diberikan list `[1, 2, 3, 4, 5]`, ubah nilai elemen ketiga dalam list menjadi `"Python"`. Cetak list sebelum dan sesudah modifikasi.

##### Soal 3: Menambahkan Elemen ke List

Diberikan sebuah list kosong, tambahkan elemen `"Python"`, `2023`, dan `[1, 2, 3]` ke dalam list tersebut menggunakan metode yang sesuai. Cetak list setelah penambahan elemen.

##### Soal 4: Menghapus Elemen dari List

Dari list `[1, 2, 3, "Python", 4, "C++", 5]`, hapus `"Python"` dan `5` dari list. Cetak list sebelum dan sesudah penghapusan.

##### Soal 5: List Slicing dan Manipulasi

Diberikan list `[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]`, lakukan operasi berikut dan cetak hasilnya:

- Ambil bagian list dari elemen kedua hingga elemen kelima.
- Ambil semua elemen dengan indeks genap.
- Balik urutan elemen dalam list.

##### Soal 6: Penggabungan dan Repetisi List

Diberikan dua list, `list1 = [1, 2, 3]` dan `list2 = ["Python", "Java", "C++"]`, gabungkan kedua list tersebut dan cetak hasilnya. Kemudian, cetak repetisi `list1` sebanyak 3 kali.

##### Soal 7: List Comprehension

Gunakan list comprehension untuk membuat list baru yang berisi kuadrat dari semua angka yang ada dalam list `[1, 2, 3, 4, 5]`. Cetak list hasilnya.

##### Soal 8: Nested List

Diberikan nested list `[[1, 2, 3], [4, 5, 6], [7, 8, 9]]`, akses dan cetak angka `5` dari list tersebut.

### SOAL TUPLE :

#### Soal 1: Membuat dan Mengakses Tuple

Buatlah sebuah tuple yang berisi beberapa jenis buah. Kemudian, akses elemen pertama dan terakhir dari tuple tersebut.

#### Soal 2: Mengubah Tuple

Tuple adalah tipe data yang immutable. Namun, bagaimana Anda dapat mengubah isi dari sebuah tuple? Buatlah sebuah contoh.

#### Soal 3: Menghitung Elemen

Diberikan sebuah tuple yang berisi nama-nama buah dengan beberapa nama buah yang muncul lebih dari satu kali. Hitunglah berapa kali "apel" muncul dalam tuple

#### Soal 4: Slicing Tuple

Diberikan sebuah tuple yang berisi angka dari 1 sampai 10. Buatlah sebuah slice dari tuple tersebut yang hanya berisi angka ganjil.

#### Soal 5: Menggabungkan Tuple

Diberikan dua tuple, gabungkan kedua tuple tersebut menjadi satu tuple baru.

#### Soal 6: Mencari Index

Diberikan sebuah tuple yang berisi beberapa elemen, carilah index dari elemen tertentu dalam tuple tersebut.

### SOAL SET :

#### Soal 1: Membuat dan Menambahkan Elemen

Buat sebuah set kosong bernama `my_set` dan tambahkan elemen `1`, `2`, dan `3` ke dalamnya. Kemudian, coba tambahkan elemen `3` lagi dan cetak `my_set` untuk melihat hasilnya. Apa yang Anda perhatikan tentang elemen duplikat?

#### Soal 2: Penghapusan Elemen

Diberikan set berikut: `numbers = {1, 2, 3, 4, 5}`



## Panduan Praktikum Pertemuan ke-2

Hapus elemen `4` dari `numbers` menggunakan metode yang sesuai. Apa yang terjadi jika Anda mencoba menghapus elemen yang tidak ada dalam set?

### Soal 3: Operasi Set

Diberikan dua set:

pythonCopy code

```
a = {1, 2, 3, 4}
```

```
b = {3, 4, 5, 6}
```

- Hitung union dari `a` dan `b`.
- Hitung intersection dari `a` dan `b`.
- Hitung difference antara `a` dan `b`.
- Hitung symmetric difference antara `a` dan `b`.

### Soal 4: Memeriksa Keanggotaan

Diberikan set berikut:

pythonCopy code : `my_fruits = {"apple", "banana", "cherry"}`

Tulis kode untuk memeriksa apakah `"apple"` ada di dalam `my_fruits`. Kemudian, coba periksa apakah `"orange"` ada di dalamnya.

### Soal 5: Operasi dengan Set Lain

Diberikan dua set:

pythonCopy code :

```
x = {1, 2, 3}
```

```
y = {4, 3, 6}
```

- Tambahkan semua elemen dari `y` ke `x`.
- Kemudian, cari elemen-elemen yang ada di kedua set tersebut (`intersection`).
- Setelah itu, hapus semua elemen yang ada di `y` dari `x`.

## SOAL DICTIONARY :

### Soal 1: Membuat dan Mengakses Dictionary

1. Buatlah sebuah dictionary yang menyimpan informasi tentang sebuah buku. Dictionary tersebut harus memiliki kunci `judul`, `pengarang`, dan `tahun`. Isilah dictionary dengan informasi buku pilihan Anda.
2. Akses dan cetak nilai `pengarang` dari dictionary yang telah Anda buat.

### Soal 2: Menambahkan dan Memodifikasi Elemen

## Panduan Praktikum Pertemuan ke-2

Diberikan dictionary berikut:

pythonCopy code :

```
karyawan = {"nama": "John Doe", "usia": 28, "departemen": "Keuangan"}
```

1. Tambahkan kunci **gaji** dengan nilai **7000** ke dalam dictionary **karyawan**.
2. Ubah nilai **departemen** menjadi **"Teknologi Informasi"**.
3. Cetak dictionary **karyawan** setelah perubahan.

### Soal 3: Menghapus Elemen

Dari dictionary **karyawan** yang sudah dimodifikasi, hapus kunci **usia** dan cetak dictionary hasilnya.

### Soal 4: Looping Melalui Dictionary

Diberikan dictionary berikut:

pythonCopy code : `nilai_mahasiswa = {"Alice": 85, "Bob": 93, "Clara": 88}`

Tulis loop yang mencetak nama dan nilai setiap mahasiswa dalam format "Nama: Nilai".

### Soal 5: Dictionary Comprehension

Gunakan dictionary comprehension untuk mengubah **nilai\_mahasiswa** menjadi dictionary baru **lulus** dimana hanya siswa dengan nilai lebih dari 90 yang dimasukkan, dengan format yang sama.

### Soal 6: Gabungkan Dua Dictionary

Diberikan dua dictionary:

pythonCopy code :

```
dict1 = {"a": 100, "b": 200}
```

```
dict2 = {"x": 300, "y": 200}
```

Gabungkan kedua dictionary tersebut menjadi satu dictionary baru dan cetak hasilnya.

--- Selamat Belajar---