

# Materi Praktikum Python: List

Tipe Data Dasar dalam Python  
Praktikum Pemrograman

# Apa itu List?

- **List** adalah salah satu tipe data terpenting di Python yang digunakan untuk menyimpan sekumpulan data dalam satu variabel.
- Elemen di dalam list dapat diubah, ditambah, atau dihapus, sehingga bersifat mutable (dinamis).
- **List** ditandai dengan tanda kurung siku [ ].
- Elemen dalam list dapat berupa tipe data apapun (angka, string, boolean, dll).

# Contoh List

Contoh kode Python:

```
buaH = ['apel', 'jeruk', 'mangga']
```

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

```
campur = ['Farid', 22, True]
```

```
print(buaH)
```

```
print(angka)
```

```
print(campur)
```

# Kegunaan List

- Menyimpan data yang jumlah atau isinya **dapat berubah**.
- Menyimpan **daftar atau koleksi data** yang berurutan.
- Cocok untuk struktur data seperti antrian, daftar nama, daftar barang, hasil perhitungan, dan sebagainya.
- Banyak digunakan dalam pengolahan data, algoritma, aplikasi, serta antarmuka pengguna.

# Kelebihan List

- Mudah dimanipulasi (ditambah, dihapus, diubah).
- Fleksibel menyimpan berbagai jenis data.
- Mendukung operasi looping dan pengindeksan.
- Cocok untuk data dinamis.

# Kekurangan List

- Karena bersifat mutable, perubahan pada list bisa berdampak pada bagian program lain yang menggunakan list tersebut.
- Lebih lambat dibanding tuple dalam beberapa operasi, karena fleksibilitasnya.

## A.Membuat List

```
Data ke- 0          Data ke- 1          Data ke- 2  
↑               ↑               ↑  
buah = ["apel", "jeruk", "mangga"]  
print(buah)
```

## B. Mengakses Elemen

```
print(buah[0])      # apel  
print(buah[2])      # mangga  
print(buah[-1])     # elemen terakhir
```

## C. Menambah Data

```
buah.append("pisang")          # tambah di akhir  
buah.insert(1, "melon")        # tambah di posisi ke-1  
buah.extend(["anggur", "nanas"]) # tambah banyak data  
print(buah)
```

## D. Mengubah Data

```
buah[0] = "semangka"
```

```
print(buah)
```

## E. Menghapus Data

```
buah.remove("jeruk")      # hapus berdasarkan nilai  
buah.pop(2)                # hapus berdasarkan indeks  
del buah[0]                  # hapus pakai del  
print(buah)
```

## F. Looping List

```
for item in buah:  
    print("Buah:", item)
```

# Operasi Dasar pada List

- Mengakses elemen: buah[0] # 'apel'
- Menambahkan elemen:  
`buah.append('pisang')`
- Menghapus elemen: `bah.remove('jeruk')`
- Mengubah elemen: `bah[1] = 'anggur'`
- Panjang list: `len(bah)`

# Fungsi dan Metode List

Fungsi/Metode	Kegunaan
append(x)	Menambah elemen ke akhir list
insert(i, x)	Menyisipkan elemen pada posisi i
remove(x)	Menghapus elemen dengan nilai x
pop(i)	Menghapus elemen pada indeks i
len(list)	Menghitung jumlah elemen
sort()	Mengurutkan list
reverse()	Membalik urutan list
count(x)	Hitung jumlah kemunculan elemen
index(x)	Cari posisi elemen

`append(x)` → Menambahkan elemen ke akhir list

```
buah = ["apel", "mangga", "pisang"]
buah.append("jeruk")
print(buah) # ['apel', 'mangga', 'pisang', 'jeruk']
```

`insert(i, x)` → Menyisipkan elemen pada posisi i

```
buah = ["apel", "pisang", "jeruk"]
buah.insert(1, "mangga")    # posisi 1
print(buah)   # ['apel', 'mangga', 'pisang', 'jeruk']
```

`remove(x)` → Menghapus elemen dengan nilai x

```
buah = ["apel", "mangga", "pisang", "jeruk"]
buah.remove("pisang")
print(buah) # ['apel', 'mangga', 'jeruk']
```

`pop(i)` → Menghapus elemen pada indeks i

```
buah = ["apel", "mangga", "pisang", "jeruk"]
buah.pop(2)    # hapus elemen indeks ke-2
print(buah)   # ['apel', 'mangga', 'jeruk']
```

`len(list)` → Menghitung jumlah elemen dalam list

```
buah = ["apel", "mangga", "jeruk"]
print(len(buah)) # 3
```

`sort()` → Mengurutkan list (secara alfabet/angka)

```
angka = [5, 2, 9, 1, 7]
angka.sort()
print(angka) # [1, 2, 5, 7, 9]
```

## reverse() → Membalik urutan list

```
buah = ["apel", "mangga", "pisang"]
buah.reverse()
print(buah) # ['pisang', 'mangga', 'apel']
```

`count(x)` → Menghitung jumlah kemunculan elemen x

```
angka = [1, 2, 2, 3, 4, 2]
print(angka.count(2)) # 3
```

`index(x)` → Mencari posisi (indeks) elemen x

```
buah = ["apel", "mangga", "pisang", "jeruk"]
print(buah.index("pisang")) # 2
```

# Perulangan pada List

- Contoh kode:

```
buah = ['apel', 'jeruk', 'mangga']
```

```
for item in buah:
```

```
    print(item)
```

- Output:

apel

jeruk

mangga

# Tuple

Dalam Python, tuple adalah **tipe data yang digunakan untuk menyimpan sekumpulan item dalam satu variabel**, mirip seperti list.

Bedanya, tuple bersifat *immutable* → artinya **tidak bisa diubah** setelah dibuat (tidak bisa menambah, menghapus, atau mengganti elemen di dalamnya).

# Ciri-ciri Tuple

1. Menggunakan tanda kurung biasa ()
2. Dapat menyimpan berbagai tipe data (angka, string, boolean, dsb).
3. Bersifat *immutable* (tidak dapat diubah).
4. Bisa diakses menggunakan **indeks** seperti list.
5. Bisa berisi **elemen duplikat**.

# Kegunaan Tuple

- Menyimpan data yang **bersifat tetap** atau tidak perlu diubah.
- Cocok untuk data konfigurasi, koordinat, hasil operasi yang tidak boleh dimodifikasi, dan pasangan data yang tetap.
- Sering digunakan untuk meningkatkan keamanan data.
- Karena tidak bisa diubah, tuple juga lebih efisien dalam penggunaan memori dan kecepatan akses.

# Kelebihan Tuple

- Lebih **aman** karena tidak bisa diubah secara tidak sengaja.
- Lebih **cepat** dibanding list dalam operasi tertentu.
- Dapat digunakan sebagai kunci dalam struktur data lain seperti dictionary (karena sifatnya tetap).
- Cocok untuk data yang bersifat konstan.

# Kekurangan Tuple

- Tidak fleksibel, karena tidak bisa dimodifikasi setelah dibuat.
- Tidak mendukung operasi manipulasi data seperti penambahan atau penghapusan elemen secara langsung.

# Contoh

```
hewan = ("Burung", "Kucing", "Anjing")
print (hewan)
```

Akses elemen tuple :

```
print(hewan[0]) # burung
print (hewan[1]) #kucing
```

# Tuple Tidak Bisa Diubah

```
hewan = ("burung", "kucing", "anjing")
```

```
hewan[1] = "ular" # ERROR – tuple tidak bisa  
diubah
```

# Menambah item baru

```
hewan = ("burung", "kucing", "anjing")
hewan_baru = hewan + ("ular",)    # menambahkan dengan membuat tuple baru
print(hewan_baru)
```

# Unpacking Tuple

menyimpan isi tuple ke dalam beberapa variabel sekaligus:

```
data = ("Farid", 20, "Indonesia")
nama, umur, negara = data

print(nama)      # Farid
print(umur)      # 20
print(negara)    # Indonesia
```

# Contoh Program: list dan tuple

```
# Daftar belanja (List berisi tuple)
daftar_belanja = [
    ("Beras", 50000),
    ("Telur", 30000),
    ("Gula", 20000),
    ("Kopi", 15000)
]

# Menampilkan daftar belanja
print("== Daftar Belanja Hari Ini ==")
total = 0
for barang, harga in daftar_belanja:
    print(f"- {barang}: Rp{harga}")
    total += harga

print("-----")
print(f"Total Belanja: Rp{total}")
```

# Tugas Mandiri

## Latihan :

- 1.Buat list berisi nama-nama teman (minimal 5 nama).
- 2.Tambahkan minimal 2 nama baru.
- 3.Hapus salah satu nama.
- 4.Tampilkan semua nama dengan looping.