

## ▼ 1. Buatlah code menggunakan Algoritma Insertion Sort dengan ketentuan sebagai berikut:

### ▼ a. Ascending tetapi proses pengurutan dimulai dari indeks-indeks terakhir

# Ascending tetapi proses pengurutan dimulai dari indeks-indeks terakhir

```
def insertionSortAsc(arr):
    n = len(arr)
    iterasi = 1
    for i in range(n - 2, -1, -1):
        key = arr[i]
        print(f"iterasi {iterasi}")
        print(f'key = {key}')
        iterasi += 1
        j = i + 1
        while j < n and key > arr[j]:
            arr[j - 1] = arr[j]
            j += 1
        arr[j - 1] = key
        print(arr)

array = [9, 5, 1, 4, 3]
print("Array sebelum diurutkan:", array)
insertionSortAsc(array)
print("Array setelah diurutkan:", array)
```

Array sebelum diurutkan: [9, 5, 1, 4, 3]  
 iterasi 1  
 key = 4  
 [9, 5, 1, 3, 4]  
 iterasi 2  
 key = 1  
 [9, 5, 1, 3, 4]  
 iterasi 3  
 key = 5  
 [9, 1, 3, 4, 5]  
 iterasi 4  
 key = 9  
 [1, 3, 4, 5, 9]  
 Array setelah diurutkan: [1, 3, 4, 5, 9]

### ▼ b. Descending tetapi proses pengurutan dimulai dari indeks-indeks terakhir

```
def insertionSortDsc(arr):
    n = len(arr)
    iterasi = 1
    for i in range(n - 2, -1, -1):
        key = arr[i]
        print(f"iterasi {iterasi}")
        print(f'key = {key}')
        iterasi += 1
        j = i + 1
        while j < n and key < arr[j]:
            arr[j - 1] = arr[j]
            j += 1
        arr[j - 1] = key
        print(arr)

array = [9, 5, 1, 4, 3]
print("Array sebelum diurutkan:", array)
insertionSortDsc(array)
print("Array setelah diurutkan:", array)
```

Array sebelum diurutkan: [9, 5, 1, 4, 3]  
 iterasi 1  
 key = 4  
 [9, 5, 1, 4, 3]  
 iterasi 2  
 key = 1  
 [9, 5, 4, 3, 1]  
 iterasi 3  
 key = 5

```
[9, 5, 4, 3, 1]
iterasi 4
key = 9
[9, 5, 4, 3, 1]
Array setelah diurutkan: [9, 5, 4, 3, 1]
```

2. Buatlah visualiasi dengan menggunakan teks, perpindahan piringan dari tower asal sampai dengan tower tujuan. Visualisasikan perpindahan piringan ini satu persatu.

```
data = {
    "A": [1, 2, 3, 4],
    "B": [],
    "C": []
}

def display(data):
    for key, value in data.items():
        print(f'{key}:')
        for item in value:
            print(f'|{item}|')

def move(asal, destinasi):
    disk = data[asal].pop(0)
    data[destinasi].insert(0, disk)

def towers(n, asal, bantuan, destinasi):
    if n == 1:
        print(f'Memindahkan lempengan 1 dari {asal} ke {destinasi}')
        move(asal, destinasi)
        display(data)
    else:
        towers(n - 1, asal, destinasi, bantuan)
        print(f'Memindahkan lempengan {n} dari {asal} ke {destinasi}')
        move(asal, destinasi)
        display(data)
        towers(n - 1, bantuan, asal, destinasi)

print('Pemindahan 4 lempengan dari A ke C dengan bantuan B')
display(data)
towers(4, 'A', 'B', 'C')
```



```
...
|4|
Memindahkan lempengan 3 dari B ke C
A:
|1|
|2|
B:
C:
|3|
|4|
Memindahkan lempengan 1 dari A ke B
A:
|2|
B:
|1|
C:
|3|
|4|
Memindahkan lempengan 2 dari A ke C
A:
B:
|1|
C:
```

[Produk berbayar Colab](#) - [Batalan kontrak di sini](#)

✓ 0 d selesai pada 21.29

