

## 1. Fungsi Unordered Sequential Search yang dapat mengoutputkan indeks indeks data kembar dan jumlah jumlah iterasi yang diperlukan

```
def customUnSeqSearch(listData, data):
    positions = []
    idx = 0
    iterations = 0
    while idx < len(listData):
        iterations += 1
        if listData[idx] == data:
            positions.append(idx)
        idx += 1
    if positions:
        return positions, iterations
    else:
        return 'Data tidak ada', iterations
```

### Hasil Running

```
data = [1, 5, 9, 8, 1, 5, 10, 26, 5, 12]
hasil, jumlahIterasi = customUnSeqSearch(data, 0)
print(f'Posisi data = {hasil}')
print(f'Jumlah iterasi = {jumlahIterasi}')
```

```
Posisi data = Data tidak ada
Jumlah iterasi = 10
```

```
data = [1, 5, 9, 8, 1, 5, 10, 26, 5, 12]
hasil, jumlahIterasi = customUnSeqSearch(data, 9)
print(f'Posisi data = {hasil}')
print(f'Jumlah iterasi = {jumlahIterasi}')
```

```
Posisi data = [2]
Jumlah iterasi = 10
```

```
data = [1, 5, 9, 8, 1, 5, 10, 26, 5, 12]
hasil, jumlahIterasi = customUnSeqSearch(data, 5)
print(f'Posisi data = {hasil}')
print(f'Jumlah iterasi = {jumlahIterasi}')
```

```
Posisi data = [1, 5, 8]
Jumlah iterasi = 10
```

## 2. Fungsi Ordered Sequential Search yang dapat mengoutputkan indeks indeks data kembar dan jumlah jumlah iterasi yang diperlukan

```
def customOrSeqSearch(listData, data):
    positions = []
    iterations = 0
    idx = 0
    stop = False
    while idx < len(listData) and not stop:
        iterations += 1
        if listData[idx] == data:
            positions.append(idx)
        else:
            if listData[idx] > data:
                stop = True
        idx += 1
    if positions:
        return positions, iterations
    else:
        return 'Data tidak ada', iterations
```

### Hasil Running

```
data = [1, 1, 5, 5, 5, 8, 9, 10, 12, 26]
hasil, jumlahIterasi = customOrSeqSearch(data, 0)
print(f"Posisi data = {hasil}")
print(f"Jumlah iterasi = {jumlahIterasi}")
```

```
Posisi data = Data tidak ada
Jumlah iterasi = 1
```

```
data = [1, 1, 5, 5, 5, 8, 9, 10, 12, 26]
hasil, jumlahIterasi = customOrSeqSearch(data, 9)
print(f"Posisi data = {hasil}")
print(f"Jumlah iterasi = {jumlahIterasi}")
```

```
Posisi data = [6]
Jumlah iterasi = 8
```

```
data = [1, 1, 5, 5, 5, 8, 9, 10, 12, 26]
hasil, jumlahIterasi = customOrSeqSearch(data, 5)
print(f"Posisi data = {hasil}")
print(f"Jumlah iterasi = {jumlahIterasi}")
```

```
Posisi data = [2, 3, 4]
Jumlah iterasi = 6
```

3. Fungsi Binary Search dapat mencari data yang sama dan mengembalikan indeks-indeks dari data yang sama serta jumlah iterasi yang diperlukan

```
def customBinarySearch(listData, data):
    first = 0
    last = len(listData) - 1
    position = []
    iterations = 0
    stop = False
    while first <= last and not stop:
        iterations += 1
        midPoint = (first + last) // 2
        if listData[midPoint] == data:
            position.append(midPoint)
            if listData[midPoint + 1] == data:
                first = midPoint + 1
            elif listData[midPoint - 1] == data:
                last = midPoint - 1
            else:
                stop = True
        else:
            if data < listData[midPoint]:
                last = midPoint - 1
            else:
                first = midPoint + 1

    if position:
        return position, iterations
    else:
        return "Data tidak ada", iterations
```

#### Hasil Running

```
data = [1, 1, 5, 5, 5, 8, 9, 10, 12, 26]
hasil, jumlahIterasi = customBinarySearch(data, 5)
print(f"Posisi data = {hasil}")
print(f"Jumlah iterations = {jumlahIterasi}")
```

```
Posisi data = [4, 2, 3]
Jumlah iterations = 4
```

```
data = [1, 1, 5, 5, 5, 8, 9, 10, 12, 26]
hasil, jumlahIterasi = customBinarySearch(data, 10)
print(f"Posisi data = {hasil}")
print(f"Jumlah iterations = {jumlahIterasi}")
```

```
Posisi data = [7]
Jumlah iterations = 2

data = [1, 1, 5, 5, 5, 8, 9, 10, 12, 26]
hasil, jumlahIterasi = customBinarySearch(data, 1)
print(f"Posisi data = {hasil}")
print(f"Jumlah iterations = {jumlahIterasi}")

Posisi data = [1, 0]
Jumlah iterations = 3

data = [1, 1, 5, 5, 5, 8, 9, 10, 12, 26]
hasil, jumlahIterasi = customBinarySearch(data, 20)
print(f"Posisi data = {hasil}")
print(f"Jumlah iterations = {jumlahIterasi}")

Posisi data = Data tidak ada
Jumlah iterations = 4
```

---

✓ 0 d selesai pada 22.44

