



## LABORATORIUM SISTEM INFORMASI UNIVERSITAS TANJUNGPURA PONTIANAK

Gedung FMIPA Jl. Prof. Dr. Hadari Nawawi Pontianak

Hari/Tanggal: Kamis / 20 Februari 2025

<b>Nama Mahasiswa</b>	Rafli Pratama	<b>Mata Kuliah Praktikum</b>	Algortima dan Struktur Data
<b>NIM</b>	H1101241008	<b>Dosen Pengampu</b>	Ilhamsyah S.Si, M.Cs
<b>Semester</b>	2 Genap	<b>Paraf Dosen Pengampu</b>	
<b>Kelas</b>	Sistem Informasi A	<b>Asisten Praktikum</b>	
<b>Nilai</b>		<b>Paraf Asisten Praktikum</b>	

### LEMBAR KERJA PRAKTIKUM

MATERI PRAKTIKUM : Searching

Percobaan 1.

```
#Sequential Search
def seq_search(nums,x):
    for i in range(len(nums)):
        if x == nums[i]:
            return i
    return -1

S = [11,37,45,26,59,28,17,53]
x = 53
pos = seq_search(S,x)
print(f'Posisi bilangan {x} didalam list S adalah posisi nomor {pos}')
```

Output Program :  
Posisi bilangan 53 didalam list S  
adalah posisi nomor 7

Penjelasan :

Pada program ini kita membuat fungsi bernama `seq_search` yang menerima parameter `nums` dan `x` dan didalamnya kita melakukan perulangan `for` sebanyak jumlah data `nums`, dan jika nilai `x` sama dengan nilai `nums` index ke-`i` maka program akan mereturn `i` dan jika tidak ada yang memenuhi sampai akhir, def akan mengembalikan nilai `-1`

Percobaan 2.

```
def seq_search(nums,x):
    for i in range(len(nums)):
        if x == nums[i]:
            print(f'Posisi bilangan {x} didalam list S adalah posisi nomor {i}')

def inpdata():
    data = []
    a = int(input('Masukkan jumlah data : '))
    for i in range(a):
        inp = int(input('Data :'))
        data.append(inp)

    print(data)
    x = int(input('Masukkan nilai yang ingin dicari'))
    return data , x

nums,x = inpdata()
seq_search(nums,x)
```

Output Program :

```
[12, 12, 14, 15]
Posisi bilangan 12 didalam list S adalah posisi nomor 0
Posisi bilangan 12 didalam list S adalah posisi nomor 1
```

Penjelasan : Pada program ini kita membuat fungsi bernama `seq_search` yang menerima parameter `nums` dan `x` dan didalamnya kita melakukan perulangan `for` sebanyak jumlah data `nums`, dan jika nilai `x` sama dengan nilai `nums` index ke-`i` maka program akan mencetak posisi dari `x`,

Dan dibawahnya kita membuat fungsi `inpdata` yang didalamnya kita membuat variabel `data` dengan isi list kosong, lalu user diminta untuk menentukan berapa banyak data yang ingin ada didalam data dengan var `a`, lalu melalui perulangan `for` sebanyak `a`, user di minta memasukkan data yang ingin di `append` ke `data`, setelah itu user akan diminta memasukkan angka yang ingin di cari, lalu fungsi akan mereturn `nums` dan `x`, dan var `nums`, dan `x` digunakan untuk menjalankan fungsi `seq_search`

### Percobaan 3.

```
#Latihan 2

def seq_search(nums,x):
    for i in range(len(nums)):
        if x == nums[i]:
            return i

    return('Nilai tidak di temukan')

def inpdata():
    a = int(input('Masukkan jumlah data : '))
    for i in range(a):
        inp = int(input('Data : '))
        data.append(inp)

    print(data)

data = []
global nums, x, pos
inpdata()
x = int(input('Masukkan nilai yang ingin dicari : '))
pos = seq_search(data,x)
print(f'Posisi bilangan {x} didalam list S adalah posisi nomor {pos}')
```

#### Output Program :

```
[12, 3, 23, 23]
Posisi bilangan 23
didalam list S adalah
posisi nomor 2
```

#### Penjelasan :

Pada program ini kita membuat fungsi bernama `seq_search` yang menerima parameter `nums` dan `x` dan didalamnya kita melakukan perulangan `for` sebanyak jumlah data `nums`, dan jika nilai `x` sama dengan nilai `nums` index ke-`i` maka program akan mereturn `i`

Dan dibawahnya kita membuat fungsi `inpdata` yang didalamnya user diminta untuk menentukan berapa banyak data yang ingin ada didalam data dengan var `a`, lalu melalui perulangan `for` sebanyak `a`, user di minta memasukkan data yang ingin di `append` ke data

Lalu kita membuat variabel data dengan isi list kosong, lalu kita menggunakan global untuk var `nums`, `x`, dan `pos` agar bisa di akses oleh `def`, dan user di minta untuk memasukkan angka yang ingin di cari lalu fungsi `seq_search` dan `inpdata` dijalankan

### Percobaan 4.

```
#Latihan 3
data=[[13,44,46,33,57],[45,67,54,32,45],[89,90,87,65,44],[23,45,67,32,10]]
print(data)
print(data[0])
print(data[2])
print(data[0][2])
print(data[2][3])

for i in data:
    for j in i :
        print(j, end = ' ')
    print()
```

#### Output Program :

```
[[13, 44, 46, 33, 57], [45, 67,
54, 32, 45], [89, 90, 87, 65, 44],
[23, 45, 67, 32, 10]]
[13, 44, 46, 33, 57]
[89, 90, 87, 65, 44]
46
65
13 44 46 33 57
45 67 54 32 45
89 90 87 65 44
23 45 67 32 10
```

#### Penjelasan :

Pada program ini kita membuat list 2 dimensi dengan 3 elemen, yang setiap elemennya memiliki 3 elemen lagi atau bisa di bilang 3 x 3, lalu kita mencetak list tersebut sesuai index

#### Percobaan 5.

```
#Latihan 4
#contoh 1
nama = ['dani', 'budi', 'lala']
nilai = ['78', '88', '92']

for i,raport in enumerate(nama):
    print(f'{raport} ---> {nilai[i]}')
```

#### Output Program :

```
dani ---> 78  
budi ---> 88  
lala ---> 92
```

#### Penjelasan :

Pada program ini kita membuat list nama dan list nilai, lalu kita melakukan perulangan for dengan fungsi enumerate yang 1 untuk menyimpan index dan raport, untuk menyimpan isi list nama, lalu didalam perulangan tersebut kita mencetak raport dan juga nilai dengan index ke i

#### Percobaan 6.

```
#Latihan 4  
#contoh 2  
  
mymatrix = [[12,24,36],[34,25,46],[27,38,49]]  
  
for i,j in enumerate(mymatrix):  
    for k,l in enumerate(j):  
        if l == 24:  
            print(i,k)
```

Output  
Program :  
0 1

#### Penjelasan :

Pada program ini kita membuat list matrix, lalu didalamnya ada list 3x3, dan kita melakukan perulangan for untuk mengakses setiap index, jika ada list dengan dengan index ke – l yang nilainya sama dengan 24, maka program akan mencetak posisinya

#### Percobaan 7.

```
#Latihan 5  
print('Program input bilangan matriks dinamis')  
bar = int(input('Jumlah baris : '))  
kol = int(input('Jumlah kolom : '))  
matriks = []  
for i in range(bar):  
    a = []  
    for j in range(kol):  
        a.append(int(input('Masukkan Data : ')))
```

```

matriks.append(a)

for i in range(bar):
    for i in range(kol):
        print(matriks[i][j], end = ' ')
    print()

```

#### Output Program :

```

Program input bilangan matriks
dinamis
1 1
1 1

```

#### Penjelasan :

Pada program ini user diminta memasukkan baris dan kolom yang menentukan list berapa kali berapa, lalu setiap nilai bar dan kolom di append ke var matriks, lalu menggunakan perulangan for untuk mengakses setiap index, dan jika ada matriks dengan index tertentu yang sama dengan nilai yang ingin dicari program akan mengembalikan posisinya

#### Percobaan 8.

```

#Latian 6
data=[[13,44,46,33,57],[45,67,54,32,45],[89,90,87,65,44],[23,45,67,32,10]]
global rows, cols

for rows in data:
    for cols in rows:
        print(cols, end=' ')
    print()

x = int(input('Masukkan nilai yang ingin dicari : '))
print()
for i,j in enumerate(data):
    for k,l in enumerate(j):
        if l==x:
            print(f'Posisi bilangan {x} didalam list data adalah posisi nomor {i,k}')
        else:
            print('data tidak ditemukan')

```

#### Output Program :

```

13 44 46 33 57

```

```
45 67 54 32 45
89 90 87 65 44
23 45 67 32 10
```

Posisi bilangan 90 didalam list data adalah  
posisi nomor (2, 1)

Penjelasan :

Pada program ini kita membuat list 2 dimensi dengan matriks 3x3, dan mengglobalkan variabel columns dan rows, lalu kita menggunakan perulangan for untuk mengakses setiap index dan mengprint elemennya sesuai index. Lalu user diminta memasukkan nilai yang ingin di cari, dan kita melakukan perulangan for dengan fungsi enumerate untuk mengakses semua index dan jika ada data pada index ke- l sama dengan x, maka program akan mengembalikan posisi index dari nilai yang sama dengan x tadi

Percobaan 9.

```
#Latihan 7
data=[[13,44,46,33,57],[45,67,54,32,45],[89,90,87,65,44],[23,45,67,32,10]]
global rows, columns, x, pos

for rows in data:
    for columns in rows:
        print(columns, end=' ')
    print()

def seq_search(data,x):
    for i,j in enumerate(data):
        for k,l in enumerate(j):
            if l == x:
                return(i,k)

    return ('Nilai tidak ditemukan')

x = int(input('Masukkan nilai yang ingin di cari : '))

pos = seq_search(data,x)
print()
print(f'Posisi bilangan {x} didalam list adalah posisi nomor {pos}')
```

Output Program :

```
13 44 46 33 57
45 67 54 32 45
89 90 87 65 44
```

```
23 45 67 32 10
```

```
Posisi bilangan 44 didalam list adalah posisi nomor (0, 1)
```

Penjelasan :

Pada program ini kita membuat list 2 dimensi dengan matriks 3x3, dan mengglobalkan variabel columns x, pos x dan rows, lalu kita menggunakan perulangan for untuk mengakses setiap index dan mengprint elemennya sesuai index. Lalu user diminta memasukkan nilai yang ingin di cari, dan kita melakukan perulangan for dengan fungsi enumerate untuk mengakses semua index dan jika ada data pada index ke- l sama dengan x, maka program akan mengembalikan posisi index dari nilai yang sama dengan x tadi

Bedanya disini perulangan dengan enumerate di buat keda;am fungsi def seq\_search yang menerima parameter data dan x

Percobaan 10.

```
#Latihan 8
def bin_search(nums,x):
    low,high = 0,len(nums)
    while low <= high:
        mid = (low + high)//2
        if nums[mid] == x:
            return mid
        elif nums[mid] > x:
            high = mid - 1
        else:
            low = mid +1

    return('tidak di temukan dalam list')

S = [11,17,26,28,37,45,53,59]
x = int(input('Masukkan bilangan yang ingin dicari : '))

pos = bin_search(S,x)
print(f'Posisi bilangan {x} didalam list S adalah posisi nomor {pos}')
```

Output Program :

```
Posisi bilangan 26 didalam list
S adalah posisi nomor 2
```

Penjelasan :

Pada program ini kita membuat fungsi bin\_search yang menerima parameter nums dan x yang didaamnya kita membuat var low dengan nilai low dan high dengan nilai sebanyak elemen nums, lalu masuk ke perulangan while low <= x, kita menentukan nilai var mid dengan menambahkan low dan high



lalu dibagi 2 dan di bulatkan kebawah, jika ada nilai nums dengan index mid sama dengan nilai x, maka fungsi akan mengembalikan nilai mid, jika tidak dan jika nums index ke mid lebih besar dari x, maka high adalah mid - 1, jika tidak ;lagi maka low adalah mid + 1, jika while low <= high sudah tidak terpenuhi, maka fungsi akan mengembalikan "tidak ditemukan dalam list", terakhir kita membuat list S, dan meminta user memasukkan nilai yang ingin di cari yang di wakili oleh var x, lalu kita membuat var pos yang menjalankan fungsi bin \_search dengan parameter S dan x, lalu program akan mencetak posisinya