



## 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>	Algoritma 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

Latihan 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`

Latihan 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`

#### Latihan 4.

```
#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 ita 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-l maka program akan mereturn i

Dan dibawahnya kita membuat fungsi inpdata yang didalamnya user diminta untuk menentukan berapa banyak data yang ingin ada dildalam 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

#### Latihan 5.

```
#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

#### Latihan 6.

```
#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 I untuk menyimpan index dan raport, untuk menyimpan isi list nama, lalu didalam perulangan tersebut kita mencetak raport dan juga nilai dengan index ke i

Latihan 7.

```
#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

Latihan 8.

```
#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

#### Latihan 9.

```

#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-1 sama dengan `x`, maka program akan mengembalikan posisi index dari nilai yang sama dengan `x` tadi

#### Latihan 10.

```
#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

Latihan 11.

```
#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 sma dengan nilai x, makafungsi 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 adlah 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 membaut var pos yang meenjalanlkn fungsi bin\_serach dengan parameter S dan x, lalu program akan mencetak posisinya

Percobaan 1.

Buatlah program untuk mencari data pada array 2 dimensi (bisa ditambahkan kode program untuk member inputan data dan ukuran array).

data array:

1 3 2

10 5 8

15 24 10

Nilai yang dicari: 24

output: data 24 berada pada posisi [2][1]

yang dicari: 2

output: data 2 berada pada posisi [0][2]

petunjuk: gunakan sequential search, karena data tidak diurutkan, terdapat 2 looping untuk proses pencarian

```
data_Set = int(input('Silahkan pilih ingin menggunakan 1. Data Default \n2. Data Custom : '))
if data_Set == 1:
    print('Data set yang tersedia adalah matriks 2 x 2 \n dengan data : ')
    data = [[1,3,2],[10,5,8],[15,24,10]]
elif data_Set==2:
    rows = int(input('Silahkan masukkan jumlah baris : '))
    cols = int(input('Silahkan masukkan jumlah kolom : '))
    print(f'Matriks yang ingin anda buat adalah matriks {rows} x {cols}')

    data = []
    for i in range(rows):
        a=[]
        for j in range(cols):
            a.append(int(input('Silahkan masukkan data')))
        data.append(a)
        print(f'Baris ke {i + 1} berhasil di buat')
    print('Data matriks : ')

    for rows in data:
        for cols in rows:
            print(cols, end = ' ')
        print()
    print()
    while True:
        element = input('Silahkan masukkan angka yang ingin di cari pada data : ("stop" untuk berhenti)').lower()
        if element == 'stop':
            print('Anda memilih berhenti')
```

```

        break
    element = int(element)
    print(f'Anda ingin mencari angka : {(element)}')
    data_ditemukan = False
    for i,rows in enumerate(data):
        for j,columns in enumerate(rows):
            if columns == element:
                print(f'Angka yang anda cari berada di posisi [{i}][{j}]')
                data_ditemukan = True

    if data_ditemukan == False:
        print('Data tidak ditemukan')

```

#### Output Program :

Jika menggunakan kondisi 1:

Data set yang tersedia adalah matriks 2 x 2  
dengan data :

```

1 3 2
10 5 8
15 24 10

```

Anda ingin mencari angka : 24

Angka yang anda cari berada di posisi [2][1]

Anda ingin mencari angka : 2

Angka yang anda cari berada di posisi [0][2]

Anda memilih berhenti

Jika menggunakan kondisi 2 :

Matriks yang ingin anda buat adalah matriks 3 x 3

Baris ke 1 berhasil di buat

Baris ke 2 berhasil di buat

Baris ke 3 berhasil di buat

Data matriks :

```

12 12 12
12 44 44
32 23 23

```

Anda ingin mencari angka : 34

Data tidak ditemukan

Anda ingin mencari angka : 23

Angka yang anda cari berada di posisi [2][1]

Angka yang anda cari berada di posisi [2][2]

Anda memilih berhenti

#### Penjelasan :

Program ini meminta user untuk memilih dataset yang ingin digunakan. Ada dua pilihan : jika memilih dataset default, maka program akan menggunakan matriks 3x3 dengan data [[1,3,2], [10,5,8], [15,24,10]]. Namun, jika memilih dataset custom, user dapat menentukan sendiri jumlah baris dan kolom

matriks yang ingin dibuat. Program kemudian meminta input angka untuk setiap elemen matriks dan menyimpannya dalam list data. Setelah matriks terbentuk, program akan mencetaknya dalam format yang rapi.

Selanjutnya program akan masuk kedalam proses pencarian angka yang dilakukan dalam perulangan while True, yang akan terus berjalan sampai user mengetikkan 'stop' jika angka ditemukan maka program akan mengembalikan index tempat angka tersebut, jika tidak maka program akan menghasilkan bahwa data tidak di temukan

#### Percobaan 2.

Buatlah program no 1 diatas dengan input fungsi input ordo matriks (baris dan kolom),, fungsi input data matriks, fungsi pencarian indeks bilangan sebuah matriks

```
def input_ordo():
    rows = int(input('Silahkan masukkan jumlah baris untuk ordo matriks : '))
    columns = int(input('Silahkan masukkan jumlah kolom untuk ordo matriks : '))
    print(f'Ordo matriks yang ingin anda buat adalah {rows} x {columns}')
    return rows, columns

def input_matriks(rows, columns):
    matriks = []
    for i in range(rows):
        a = []
        for j in range(columns):
            a.append(int(input('Silahkan masukkan data : ')))
        matriks.append(a)
        print(f'Element matriks untuk baris ke {i + 1} berhasil di buat')
    print('semua elemen matriks berhasil di input')
    return matriks

def print_matriks (matriks):
    print('Data matriks : ')
    for rows in matriks:
        for columns in rows:
            print(columns, end=' ')
        print()
    return matriks

def find_matriks(matriks):
    while True:
        element = input('Silahkan masukkan angka yang ingin di cari di dalam data :  
("Stop" untuk berhenti!')).lower()
        if element == 'stop':
            break
        element = int(element)
        print(f'Angka yang ingin anda cari adalah : {element}')
        data_ditemukan = False
```

```

for i, rows in enumerate(matriks):
    for j, columns in enumerate(rows):
        if columns == element:
            print('Data berhasil di temukan!')
            print(f'Data yang anda cari berada di posisi [{i}][{j}]')
            data_ditemukan = True

if data_ditemukan == False:
    print('Data tidak di temukan')

rows,columns = input_ordo()
find_matriks(print_matriks(input_matriks(rows,columns)))

```

#### Output Program :

Ordo matriks yang ingin anda buat adalah 3 x 3  
 Element matriks untuk baris ke 1 berhasil di buat  
 Element matriks untuk baris ke 2 berhasil di buat  
 Element matriks untuk baris ke 3 berhasil di buat  
 semua elemen matriks berhasil di input  
 Data matriks :  
 12 24 32  
 12 14 12  
 23 12 3  
 Angka yang ingin anda cari adalah : 12  
 Data berhasil di temukan!  
 Data yang anda cari berada di posisi [0][0]  
 Data berhasil di temukan!  
 Data yang anda cari berada di posisi [1][0]  
 Data berhasil di temukan!  
 Data yang anda cari berada di posisi [1][2]  
 Data berhasil di temukan!  
 Data yang anda cari berada di posisi [2][1]  
 Angka yang ingin anda cari adalah : 3  
 Data berhasil di temukan!  
 Data yang anda cari berada di posisi [2][2]

#### Penjelasan :

Pada program ini kita membuat 4 fungsi input\_ordo(), input\_matriks(), print\_matriks(), dan find\_matriks()., yang digunakan untuk menginput ordo dan data matriks, mencetak dan menemukan angka didalam matriks, pada fungsi input\_ordo(), kita meminta user untuk memasukkan jumlah baris dan kolom, lalu mencetak informasi berupa matriks berapa kali berapa yang ingin di buat

Pada fungsi input\_matriks() yang menerima parameter rows, dan collumn program akan memasuki perulangan for bersarang yang di sesuaikan dengan range row dan collums yang didalamnya user di minta memasukkan data yang nantinya akan di append ke var data yang menyimpan list

Selanjutnya, program akan mencetak matriks dengan menggunakan fungsi `print_matriks(matriks)`. Fungsi ini hanya bertugas menampilkan isi matriks dengan format yang rapi agar mudah dibaca. Terakhir, program masuk ke fungsi `find_matriks(matriks)`, yang digunakan untuk mencari elemen dalam matriks. Di dalam fungsi ini, program akan meminta user untuk memasukkan angka yang ingin dicari, dan akan terus berjalan dalam loop `while True` sampai user mengetik "stop". Jika angka yang dimasukkan ditemukan dalam matriks, program akan menampilkan pesan bahwa angka tersebut ditemukan beserta posisi indeksinya dalam format `[baris][kolom]`. Jika angka tidak ditemukan, program akan memberi tahu bahwa data tidak tersedia. Jika user mengetik "stop", maka program akan berhenti.

### Percobaan 3.

Tambah potongan fungsi program pada latihan 8, khusus untuk menambahkan data yang diinput oleh user dan gunakan fungsi sorting untuk mengurutkannya agar dapat dicari menggunakan binary search!

```
def bin_search(data,x):
    low = 0
    high = len(data) - 1

    while low <= high:
        mid = (low + high) // 2
        if data[mid] == x:
            print(f'Data berhasil di temukan!')
            return mid
        elif data[mid] < x:
            low = mid + 1
        else : high = mid - 1

    return 'Data tidak di temukan!'

def input_data():
    print('Silahkan masukkan data : ')
    data= []
    while True:
        element = input('Silahkan masukkan angka untuk element data : ("stop" untuk
berhenti) ').lower()
        if element == 'stop' :
            break
        element = int(element)
        data.append(element)
        print(f'{element} berhasil di tambahkan')
    return data

def data_sorting(data):
    data.sort()
    return data

def main():
```

```

print('Program mencari data menggunakan binary search')
data = data_sorting(input_data())
print(data)

while True:
    x = input('Silahkan masukkan angka yang ingin di cari di dalam data : ("stop"
untuk berhenti ) ').lower()
    if x == 'stop':
        print('Anda memilih keluar')
        break
    x = int(x)
    pos = bin_search(data,x)
    print(f'Angka {x} berada pada posisi {pos}')

main()

```

#### Output Program :

Program mencari data menggunakan binary search

Silahkan masukkan data :

12 berhasil di tambahkan

23 berhasil di tambahkan

43 berhasil di tambahkan

23 berhasil di tambahkan

12 berhasil di tambahkan

23 berhasil di tambahkan

24 berhasil di tambahkan

53 berhasil di tambahkan

23 berhasil di tambahkan

11 berhasil di tambahkan

12 berhasil di tambahkan

[11, 12, 12, 12, 23, 23, 23, 23, 24, 43, 53]

Data berhasil di temukan!

Angka 43 berada pada posisi 9

Angka 10 berada pada posisi Data tidak di temukan!

Data berhasil di temukan!

Angka 12 berada pada posisi 2

Data berhasil di temukan!

Angka 11 berada pada posisi 0

Anda memilih keluar

#### Penjelasan :

Program ini menggunakan binary search untuk mencari data. Terdapat 4 fungsi yang dibuat

Pertama, fungsi input\_data() digunakan untuk meminta user memasukkan angka-angka ke dalam list data. User bisa terus memasukkan angka hingga mereka mengetik "stop", yang akan mengakhiri proses input. Setiap angka yang dimasukkan akan langsung ditambahkan ke dalam list, dan program akan mencetak pesan bahwa angka tersebut berhasil ditambahkan. Setelah semua data diinputkan, fungsi ini mengembalikan list data.

Setelah data diinputkan, fungsi data\_sorting(data) akan dijalankan untuk mengurutkan angka-angka dalam list secara ascending (dari kecil ke besar) menggunakan metode .sort(). Karena Binary Search hanya

bekerja pada data yang sudah terurut, fungsi ini memastikan bahwa pencarian nanti bisa dilakukan dengan benar.

Selanjutnya, program masuk ke fungsi `bin_search(data, x)`, yang merupakan implementasi dari algoritma Binary Search. Fungsi ini mencari angka `x` dalam list data dengan menggunakan pendekatan `divide and conquer`. Cara kerjanya, program menentukan `low` (batas bawah) dan `high` (batas atas), lalu menghitung `mid` (titik tengah). Jika angka yang dicari (`x`) sama dengan angka di posisi `mid`, maka angka ditemukan, dan program langsung mengembalikan indeksinya. Jika `x` lebih besar dari `data[mid]`, berarti `x` hanya bisa ada di bagian kanan, sehingga `low` akan digeser ke `mid + 1`. Jika `x` lebih kecil, berarti pencarian dilanjutkan ke bagian kiri dengan menggeser `high` ke `mid - 1`. Jika seluruh elemen sudah diperiksa dan `x` tidak ditemukan, program akan mengembalikan pesan "Data tidak ditemukan!".

Terakhir, fungsi `main()` mengatur jalannya seluruh program. Pertama, program mencetak pesan bahwa Binary Search akan digunakan. Lalu, fungsi `input_data()` dipanggil untuk mengumpulkan angka dari user, kemudian angka tersebut diurutkan dengan `data_sorting()`, dan list yang sudah diurutkan ditampilkan ke user. Setelah itu, program masuk ke loop pencarian, di mana user bisa terus memasukkan angka yang ingin dicari dalam data. Jika angka ditemukan, program akan mencetak indeksinya dalam list. Jika angka tidak ada, program akan menampilkan pesan bahwa data tidak ditemukan. Loop akan terus berjalan hingga user mengetik "stop", yang akan mengakhiri program.