


<b>Nama:</b> M.Satria Pratama  <b>NIM:</b> 065002200017	 <b>Praktikum Algoritma &amp; Pemrograman</b>	<b>MODUL 10</b>  <b>Nama Dosen:</b> Anung B. Ariwibowo, M. Kom
<b>Hari/Tanggal:</b> Jum'at, 25-11-2022		<b>Nama Asisten Labratorium:</b> 1. Azhar Rizki Zulma 065001900001

## Search, List & Sorting

### 1. Teori Singkat Linear

#### Search

Linear Search adalah sebuah algoritma pencarian, juga dikenal sebagai pencarian sekuensial, yang cocok untuk mencari sebuah nilai tertentu pada sebuah himpunan data. Algoritma ini beroperasi dengan memeriksa setiap elemen dari sebuah list sampai sebuah kecocokan ditemukan.

#### Binary Search

Binary Search atau sering disebut algoritma pencarian biner adalah sebuah teknik untuk menemukan nilai tertentu dalam sebuah larik linear, dengan menghilangkan setengah data pada setiap langkah, dipakai secara luas tetapi tidak secara eksklusif dalam ilmu komputer. Pada saat menggunakan binary search, data yang berada di dalam array harus diurutkan terlebih dahulu.

#### List

List adalah tipe data yang paling serbaguna yang tersedia dalam bahasa Python, yang dapat ditulis sebagai daftar nilai yang dipisahkan koma (item) antara tanda kurung siku. Hal penting tentang daftar adalah item dalam list tidak boleh sama jenisnya.



## Sorting

Sorting merupakan suatu proses untuk menyusun kembali humpunan obyek menggunakan aturan tertentu. Sorting disebut juga sebagai suatu algoritma untuk meletakkan kumpulan elemen data kedalam urutan tertentu berdasarkan satu atau beberapa kunci dalam tiap-tiap elemen.

Metode-metode sorting meliputi:

1. Insertion Sort (Metode Penyisipan)
2. Selection Sort (Metode Seleksi)
3. Bubble sort (Metode Gelembung)
4. Shell Sort (Metode Shell)
5. Quick Sort (Metode Quick)
6. Merge Sort (Metode Penggabungan)

Contoh pembuatan list



```
list1 = ['kimia', 'fisika', 1993, 2017]
list2 = [1, 2, 3, 4, 5]
list3 = ["a", "b", "c", "d"]
```



## 2. Alat dan Bahan Hardware :

Laptop/PC

Software : Spyder (Anaconda Python)

## 3. Elemen Kompetensi

### a. Latihan pertama

Buatlah sebuah fungsi binary search untuk mencari sebuah element didalam sebuah list tersebut yang dimana, jika list tersebut acak maka diurutkan terlebih dahulu dengan menggunakan fungsi sorting (implementasi bebas, boleh menggunakan bubblesort, dll) dan setelahnya baru dicari menggunakan fungsi binary search.

#### Source Code

```
data_list = [2, 10, 15, 23, 28, 31, 34, 56, 89, 200]

print(data_list)
cari = int(input("Masukkan Angka Yang Anda Cari : "))

def bubble_sort(List,search):
    counter = 0
    while counter != len(data_list):
        if data_list[counter] == search:
            result = counter
            return result
        counter += 1

hasil = bubble_sort(data_list,cari)
if cari not in data_list:
    print(data_list)
    print("Element tidak ditemukan pada list")
else:
    print(data_list)
    print("Element ditemukan pada posisi ke",hasil+1)
```

#### Output



```

IPdb [1]: runfile('C:/Users/satria/untitled0.py', wdir='C:/Users/satria')
[2, 10, 15, 23, 28, 31, 34, 56, 89, 200]
Masukkan Angka Yang Anda Cari : 11
[2, 10, 15, 23, 28, 31, 34, 56, 89, 200]
Element tidak ditemukan pada list

IPdb [2]: runfile('C:/Users/satria/untitled0.py', wdir='C:/Users/satria')
[2, 10, 15, 23, 28, 31, 34, 56, 89, 200]
Masukkan Angka Yang Anda Cari : 23
[2, 10, 15, 23, 28, 31, 34, 56, 89, 200]
Element ditemukan pada posisi ke 4

IPdb [3]: runfile('C:/Users/satria/untitled0.py', wdir='C:/Users/satria')
[2, 10, 15, 23, 28, 31, 34, 56, 89, 200]
Masukkan Angka Yang Anda Cari : 4
[2, 10, 15, 23, 28, 31, 34, 56, 89, 200]
Element tidak ditemukan pada list

```

b. Latihan Kedua

Buatlah sebuah fungsi sorting berdasarkan metode bubble sort menggunakan konsep rekursif dengan bahasa pemrograman Python.

Source Code

```

def data_list(n=0, data=[], i=1):
    if (n == 0):
        return 0 else
    :
        isi = int(input("Masukkan angka ke-" + str(i) + " : "))
        data.append(isi) if (n
            == 1):
                return data
    else:
        i += 1
        return data_list(n-1, data, i)

def B_sort(angka):
    for i in range(len(angka)):

```



```
for n in range(0, len(angka) - i - 1):  
    if angka[n] > angka[n + 1]:  
        x = angka[n]  
        angka[n] = angka[n+1]  
        angka[n+1] = x  
  
jumlah = int(input("Berapa banyak angka yang ingin di list : "))  
angka = data list(n = jumlah) print("\nSebelum disortir : ")  
print(angka)  
B sort(angka) print("\nSetelah disortir :  
")  
print(angka)
```

### Output



```
4.3
Berapa banyak angka yang ingin di list : 5
Masukkan angka ke-1 : 32
Masukkan angka ke-2 : 34
Masukkan angka ke-3 : 4
Masukkan angka ke-4 : 1
Masukkan angka ke-5 : 7

Sebelum disortir :
[32, 34, 4, 1, 7]

Setelah disortir :
[1, 4, 7, 32, 34]
```

#### 4. File Praktikum

Github Repository:

```
print("Taruh link github repository praktikum kalian disini yaa...")
```

#### 5. Soal Latihan Soal:

1. Mengapa dalam algoritma pencarian binary search himpunan datanya harus diurutkan terlebih dahulu? Jelaskan alasannya!
2. Deskripsikan serta narasikan jalannya alur source code program yang sebelumnya telah kalian buat pada Elemen Kompetensi Latihan Kedua!

Jawaban:

1. Karena binary search ialah suatu metode pencarian suatu data yang ada di dalam kumpulan data dimana data-data tersebut sudah berurut, jika tidak maka proses pencarian tidak akan terjadi jika data-data yang berada di dalam kumpulan data tersebut tidak berurut. Jika dilihat dari rumusnya dimana posisi awal + akhir dibagi 2 untuk mencari data tengahnya jika nilai data yang dicari lebih besar dari nilai data yang paling tengah maka kumpulan data akan dibagi menjadi dua dan data yang berada di posisi paling tengah menjadi posisi awal pencarian data dan akhirnya data pun dapat ditemukan.
2. Pertama-tama kita membuat input untuk memasukkan data dengan berjalan di fungsi dengan menambahkan data-data dari input tersebut yang awalnya list tersebut kosong,



lalu kita membuat fungsi bubble sort yang dapat mengurutkan data acak dari fungsi data\_list yang sudah dibuat sebelumnya.

## 6. Kesimpulan

- a. Dalam pengerjaan program dengan bahasa pemrograman Python, kita harus benar-benar teliti dalam menginputkan suatu fungsi untuk menampilkan suatu keluaran pada layar dengan sesuai.
- a. Kita dapat mengetahui... Bagaimana cara kerja sortir data seperti halnya bubble sort, selection sort, dll. Berbeda dengan metode .sort pada list, ini kita merancang sendiri algoritmanya dengan menyusun satu-satu seperti bubble sort dengan mengurutkan bilangan terkecil hingga terbesar layaknya gelembung di dalam suatu fungsi def yang tentunya kita buat terlebih dahulu.
- b.

## 7. Cek List (✓)

No	Elemen Kompetensi	Penyelesaian	
		Selesai	Tidak Selesai
1.	Latihan Pertama	(✓)	
2.	Latihan Kedua	(✓)	

## 8. Formulir Umpan Balik

No	Elemen Kompetensi	Waktu Pengerjaan	Kriteria
1.	Latihan Pertama	15 Menit	Baik
2.	Latihan Kedua	15 Menit	Baik

Keterangan: 1.

Menarik

2. Baik

3. Cukup

4. Kurang

