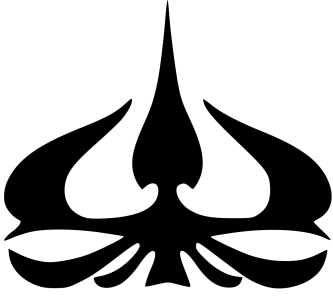


| | | |
|---|---|---|
| Nama : Rafael Gala Herlambang NIM : 064002300036 |  Algoritma dan Pemrograman Dasar | Modul 12 Nama Dosen: Abdul Rochman |
| Hari/Tanggal: Rabu, 13 Desember 2023 | | Nama Aslab: 1. Gagah Putra Bangsa (064002100036) 2. Nathanael Widjaya (064002100020) |

MODUL 12 : SEARCHING & SORTING

Deskripsi Modul : Membuat program dengan menggunakan algoritma Searching & Sorting

| No. | Elemen Kompetensi | Indikator Kinerja | Halaman |
|-----|--|--|---------|
| 1. | Mampu memahami dan menguasai algoritma searching & sorting | Memahami dan menguasai algoritma searching & sorting | 2 |

TEORI SINGKAT

Searching dalam Python adalah proses menemukan keberadaan elemen tertentu dalam suatu kumpulan data. Python menyediakan beberapa metode pencarian, termasuk Linear Search dan Binary Search. Linear Search melakukan pencarian berurutan dari awal hingga akhir kumpulan data, sementara Binary Search hanya dapat digunakan pada data yang sudah terurut dan membagi kumpulan data menjadi setengah pada setiap iterasi.

Sorting dalam Python mencakup pengaturan elemen dalam suatu kumpulan data dalam urutan tertentu. Python menyediakan berbagai algoritma sorting, seperti Bubble Sort, Selection Sort, Merge Sort, dan Quick Sort. Setiap algoritma memiliki keunggulan dan kelemahannya sendiri. Penggunaan algoritma sorting yang tepat sangat tergantung pada kebutuhan aplikasi dan sifat data yang dihadapi. Python juga menyediakan fungsi bawaan seperti `sorted()` untuk melakukan pengurutan dengan mudah pada berbagai struktur data. Dengan memahami dan memilih metode searching dan sorting yang sesuai, pengembang Python dapat meningkatkan efisiensi dan kinerja aplikasi mereka.

DAFTAR PERTANYAAN

1. Jelaskan urutan dari algoritma Linear Search!

Jawaban :

Inisialisasi: Mulai dengan inisialisasi variabel untuk menyimpan status pencarian atau indeks.

Looping: Iterasi melalui setiap elemen dalam kumpulan data.

Pemeriksaan: Periksa setiap elemen apakah sama dengan elemen yang dicari.

Pertimbangan: Jika elemen ditemukan, catat indeks atau status pencarian sebagai "ditemukan."

Pencarian Selesai: Jika seluruh kumpulan data diperiksa dan elemen tidak ditemukan, set status pencarian sebagai "tidak ditemukan."

Output Hasil: Tampilkan hasil berdasarkan status pencarian, seperti indeks ditemukan atau pesan elemen tidak ditemukan.

2. Apa saja kelebihan dari algoritma Binary Search?

Jawaban :

Algoritma Binary Search memiliki beberapa kelebihan yang membuatnya efisien dalam pencarian elemen dalam kumpulan data terurut. Dengan membagi setengah kumpulan data pada setiap iterasi, Binary Search memiliki kompleksitas waktu logaritmik, sehingga lebih cepat dibandingkan dengan Linear Search, terutama pada data yang besar. Keefektifannya muncul khususnya ketika beroperasi pada data yang sudah terurut, dan penggunaan memori yang rendah menjadikannya pilihan optimal. Binary Search juga dapat dioptimalkan untuk digunakan pada struktur data tertentu, seperti array, dan kemampuannya untuk langsung membandingkan elemen tengah setiap iterasi mengurangi jumlah langkah pencarian. Ini membuat Binary Search menjadi algoritma yang efisien dan cepat dalam menemukan elemen dalam kumpulan data yang besar.

3. Menurut Anda untuk algoritma sorting lebih cepat dan efisien Bubble Sort atau Selection Sort? Jelaskan!

Jawaban :

secara umum, Selection Sort cenderung lebih efisien dibandingkan Bubble Sort dalam konteks kecepatan dan kinerja. Bubble Sort bekerja dengan membandingkan dan menukar elemen secara berurutan, menggeser elemen terbesar ke ujung setiap iterasi.

LAB SETUP

Hal yang harus disiapkan dan dilakukan oleh praktikan untuk menjalankan praktikum modul ini, antara lain:

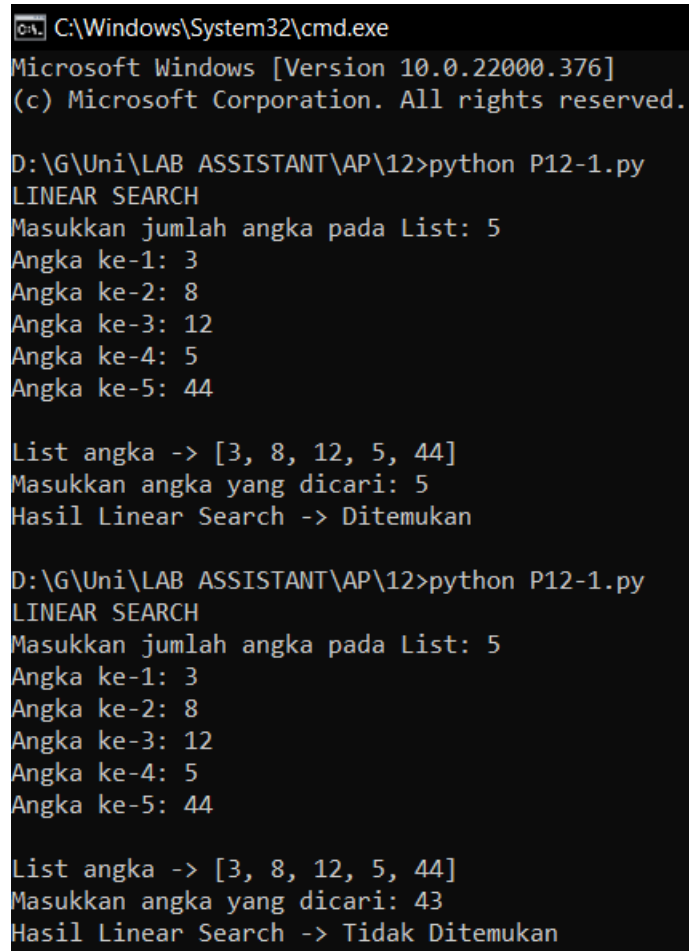
1. Menyiapkan IDE untuk membangun program python (Spyder, Sublime, dll);
2. Python sudah terinstal dan dapat berjalan dengan baik di laptop masing-masing;
3. Menyimpan semua dokumentasi hasil praktikum pada laporan yang sudah disediakan.

ELEMEN KOMPETENSI I

Deskripsi : Membuat program untuk mencari angka pada list menggunakan algoritma Linear Search

Kompetensi Dasar : Memahami bagaimana membuat program untuk mencari angka pada list menggunakan algoritma Linear Search

1. Buatlah sebuah program untuk mencari angka pada list menggunakan algoritma Linear Search, dimana List dan angka yang dicari berasal dari input user, berikut contoh running program



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.376]
(c) Microsoft Corporation. All rights reserved.

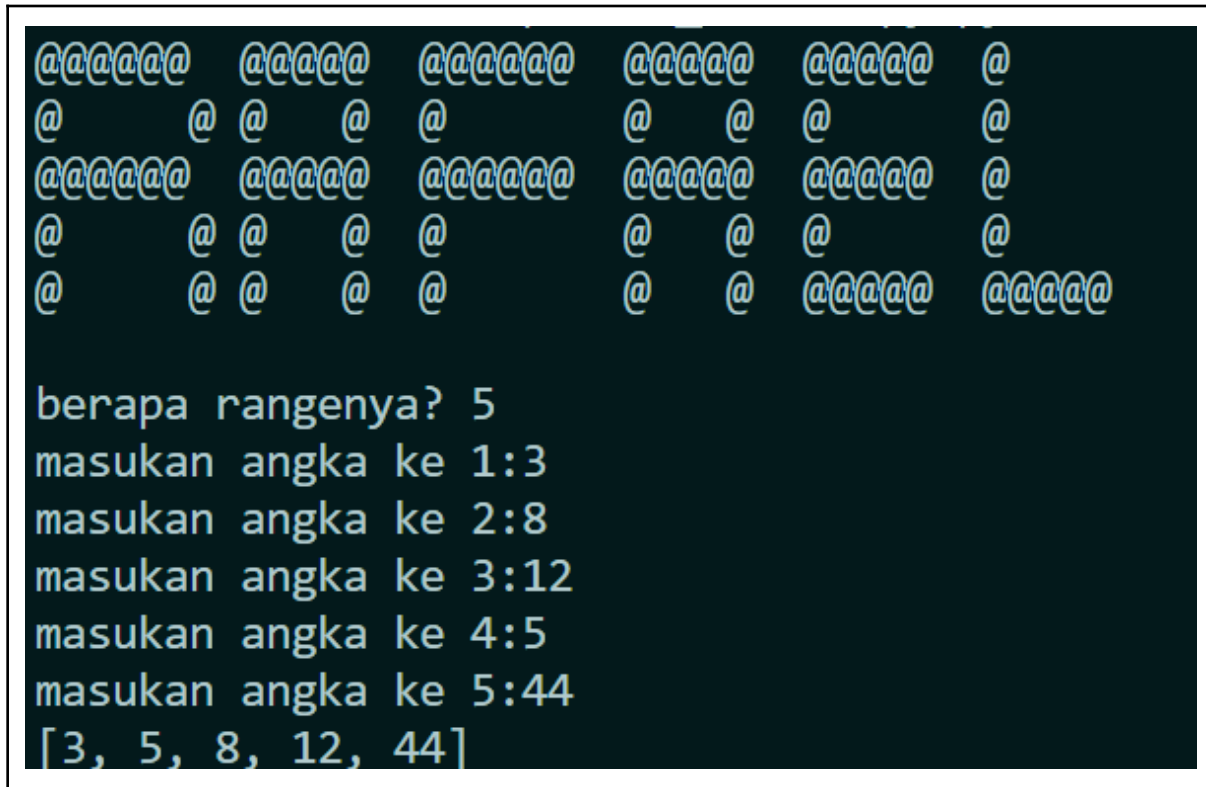
D:\G\Uni\LAB ASSISTANT\AP\12>python P12-1.py
LINEAR SEARCH
Masukkan jumlah angka pada List: 5
Angka ke-1: 3
Angka ke-2: 8
Angka ke-3: 12
Angka ke-4: 5
Angka ke-5: 44

List angka -> [3, 8, 12, 5, 44]
Masukkan angka yang dicari: 5
Hasil Linear Search -> Ditemukan

D:\G\Uni\LAB ASSISTANT\AP\12>python P12-1.py
LINEAR SEARCH
Masukkan jumlah angka pada List: 5
Angka ke-1: 3
Angka ke-2: 8
Angka ke-3: 12
Angka ke-4: 5
Angka ke-5: 44

List angka -> [3, 8, 12, 5, 44]
Masukkan angka yang dicari: 43
Hasil Linear Search -> Tidak Ditemukan
```

2. Cantumkan hasil running program.
Screenshot



ELEMEN KOMPETENSI II

Deskripsi : Membuat program untuk mencari angka pada list menggunakan algoritma Binary Search

Kompetensi Dasar : Memahami bagaimana membuat program untuk mencari angka pada list menggunakan algoritma Binary Search

1. Buatlah sebuah program untuk mencari angka pada list menggunakan algoritma Binary Search, dimana List dan angka yang dicari berasal dari input user, berikut contoh running program

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.376]
(c) Microsoft Corporation. All rights reserved.

D:\G\Uni\LAB ASSISTANT\AP\12>python P12-2.py
BINARY SEARCH
Masukkan jumlah angka pada List: 4
Angka ke-1: 3
Angka ke-2: 7
Angka ke-3: 11
Angka ke-4: 2

List angka -> [3, 7, 11, 2]
Masukkan angka yang dicari: 7
Hasil Binary Search -> Ditemukan ( di index 1 )

D:\G\Uni\LAB ASSISTANT\AP\12>python P12-2.py
BINARY SEARCH
Masukkan jumlah angka pada List: 4
Angka ke-1: 3
Angka ke-2: 7
Angka ke-3: 11
Angka ke-4: 2

List angka -> [3, 7, 11, 2]
Masukkan angka yang dicari: 8
Hasil Binary Search -> Tidak Ditemukan
```

2. Cantumkan hasil running program.
Screenshot

```

@@@@@@ @@@@@ @@@@@@@ @@@@@ @@@@@ @
@      @ @    @  @      @    @    @    @
@@@@@@ @@@@@ @@@@@@@ @@@@@ @@@@@ @
@      @ @    @  @      @    @    @    @
@      @ @    @  @      @    @@@@@ @@@@@

```

```

berapa rangenya? 4
masukan angka ke 1:3
masukan angka ke 2:7
masukan angka ke 3:11
masukan angka ke 4:2
[3, 7, 11, 2]
angka berapa?8
Angka tidak ditemukan

```

```

@@@@@@ @@@@@ @@@@@@@ @@@@@ @@@@@ @
@      @ @    @  @      @    @    @    @
@@@@@@ @@@@@ @@@@@@@ @@@@@ @@@@@ @
@      @ @    @  @      @    @    @    @
@      @ @    @  @      @    @@@@@ @@@@@

```

```

berapa rangenya? 4
masukan angka ke 1:3
masukan angka ke 2:7
masukan angka ke 3:11
masukan angka ke 4:2
[3, 7, 11, 2]
angka berapa?11
Angka ditemukan pada index ke 2

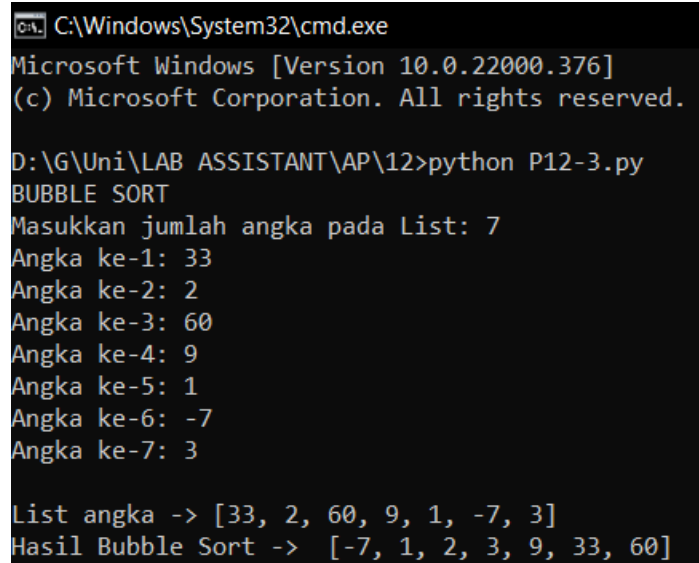
```

ELEMEN KOMPETENSI III

Deskripsi : Membuat program untuk mengurutkan list angka menggunakan algoritma Bubble Sort

Kompetensi Dasar : Memahami bagaimana membuat program untuk mengurutkan list angka menggunakan algoritma Bubble Sort

1. Buatlah sebuah program untuk mengurutkan list menggunakan algoritma Bubble Sort, dimana List berasal dari input user, berikut contoh running program



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.376]
(c) Microsoft Corporation. All rights reserved.

D:\G\Uni\LAB ASSISTANT\AP\12>python P12-3.py
BUBBLE SORT
Masukkan jumlah angka pada List: 7
Angka ke-1: 33
Angka ke-2: 2
Angka ke-3: 60
Angka ke-4: 9
Angka ke-5: 1
Angka ke-6: -7
Angka ke-7: 3

List angka -> [33, 2, 60, 9, 1, -7, 3]
Hasil Bubble Sort -> [-7, 1, 2, 3, 9, 33, 60]
```

2. Cantumkan hasil running program.
Screenshot

```
##### @
@      @ @      @      @      @      @
##### @##### @##### @##### @##### @
@      @ @      @      @      @      @
@      @ @      @      @      @      @##### @#####

berapa rangenya? 7
masukan angka ke 1:-1
masukan angka ke 2:90
masukan angka ke 3:86
masukan angka ke 4:54
masukan angka ke 5:32
masukan angka ke 6:2
masukan angka ke 7:7
list normalnya: [-1, 90, 86, 54, 32, 2, 7]
Hasil Bubble Sort = [-1, 2, 7, 32, 54, 86, 90]
```

ELEMEN KOMPETENSI IV

Deskripsi : Membuat program untuk mengurutkan list angka menggunakan algoritma Selection Sort

Kompetensi Dasar : Memahami bagaimana membuat program untuk mengurutkan list angka menggunakan algoritma Selection Sort

1. Buatlah sebuah program untuk mengurutkan list menggunakan algoritma Selection Sort, dimana List berasal dari input user, berikut contoh running program

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.376]
(c) Microsoft Corporation. All rights reserved.

D:\G\Uni\LAB ASSISTANT\AP\12>python P12-4.py
SELECTION SORT
Masukkan jumlah angka pada List: 4
Angka ke-1: 22
Angka ke-2: 87
Angka ke-3: 54
Angka ke-4: 11

List angka -> [22, 87, 54, 11]
Hasil Selection Sort -> [11, 22, 54, 87]
```

2. Cantumkan hasil running program.

Screenshot

```
@@@@@@ @@@@@ @@@@@@@ @@@@@ @@@@@ @
@      @ @      @      @      @      @
@@@@@@ @@@@@ @@@@@@@ @@@@@ @@@@@ @
@      @ @      @      @      @      @
@      @ @      @      @      @@@@@ @@@@@

Berapa kali maunya? 4
Masukan angka ke 1: 78
Masukan angka ke 2: 77
Masukan angka ke 3: 60
Masukan angka ke 4: 23
list: [78, 77, 60, 23]
list dengan selection: [23, 60, 77, 78]
```

Link Github :

Isi sendiri

KESIMPULAN (*minimal 3 baris*)

Dalam "Membuat program dengan menggunakan algoritma Searching & Sorting," kita mengimplementasikan dua aspek krusial dalam pengembangan perangkat lunak, yaitu pencarian dan pengurutan data. Algoritma searching, seperti Linear Search, membantu menemukan keberadaan suatu elemen dalam kumpulan data, sementara algoritma sorting, seperti Bubble Sort atau Selection Sort, berfokus pada pengurutan elemen dalam suatu urutan tertentu. Ini memberikan pemahaman praktis tentang implementasi algoritma ini dalam pemrograman, menunjukkan kelebihan dan kekurangannya. Siswa dapat mengambil manfaat dalam memahami bagaimana memilih algoritma yang tepat tergantung pada skenario dan kebutuhan aplikasi, serta pentingnya efisiensi dan kinerja dalam pengelolaan data.

CEKLIST

- | | |
|---|-----|
| 1. Membuat program mencari angka dengan Linear Search | (✓) |
| 2. Membuat program mencari angka dengan Binary Search | (✓) |
| 3. Membuat program mengurutkan list dengan Bubble Sort | (✓) |
| 4. Membuat program mengurutkan list dengan Selection Sort | (✓) |