|  |  |  |
| --- | --- | --- |
| **Nama:**  **(Isi Nama Anda)**  **NIM: (Isi NIM Anda)** | C:\Users\RPL-SI 02\Pictures\288px-Trisakti_Logo.svg.png | **MODUL 10**  **Nama Dosen:**  **Anung B. Ariwibowo, M. Kom** |
| **Hari/Tanggal:**  **Hari, Tanggal Bulan 2022** | **Praktikum Algoritma & Pemrograman** | **Nama Asisten Labratorium:**   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 ekslusif 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

|  |
| --- |
|  |

1. **Alat dan Bahan**

Hardware : Laptop/PC

Software : Spyder (Anaconda Python)

1. **Elemen Kompetensi**
   1. 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

|  |
| --- |
| print(“Isi dengan Source Code kalian ya…”) |

Output

|  |
| --- |
| print(“Screenshot hasil keluaran atau output dari source code kalian ya…”) |

* 1. Latihan Kedua

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

Source Code

|  |
| --- |
| print(“Isi dengan Source Code kalian ya…”) |

Output

|  |
| --- |
| print(“Screenshot hasil keluaran atau output dari source code kalian ya…”) |

1. **File Praktikum**

Github Repository:

|  |
| --- |
| print(“Taruh link github repository praktikum kalian disini yaa…”) |

1. **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. (Isi Dengan Jawaban Kalian)
2. (Isi Dengan Jawaban Kalian)
3. **Kesimpulan**
   1. Dalam pengerjaan program dengan bahasa pemrograman Python, kita harus benar-benar teliti dalam menginputkan suatu fungsi untuk menampilkan suatu keluaran pada layar dengan sesuai.
   2. Kita dapat mengetahui… (Tolong Isi lebih dari dua baris!)
4. **Cek List (✓)**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Elemen Kompetensi** | **Penyelesaian** | |
| **Selesai** | **Tidak Selesai** |
| **1.** | Latihan Pertama |  |  |
| **2.** | Latihan Kedua |  |  |

1. **Formulir Umpan Balik**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Elemen Kompetensi** | **Waktu Pengerjaan** | **Kriteria** |
| **1.** | Latihan Pertama | … Menit | … |
| **2.** | Latihan Kedua | … Menit | … |

Keterangan:

1. Menarik
2. Baik
3. Cukup
4. Kurang