


<p>Nama: Aini Rihhadatul Aisy</p> <p>NIM: 064102400024</p>	 <p>Praktikum Algoritma & Pemrograman</p>	<p>MODUL 10</p> <p>Nama Dosen: Binti solihah, S.T, M.KOM</p>
<p>Hari/Tanggal: Jumat, 29 November 2024</p>		<p>Nama Asisten Labratorium:</p> <ol style="list-style-type: none"> 1. Yustianas Rombon - 064002300015 2. Vira Aditya Kurniawan - 065002300012

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"]
```

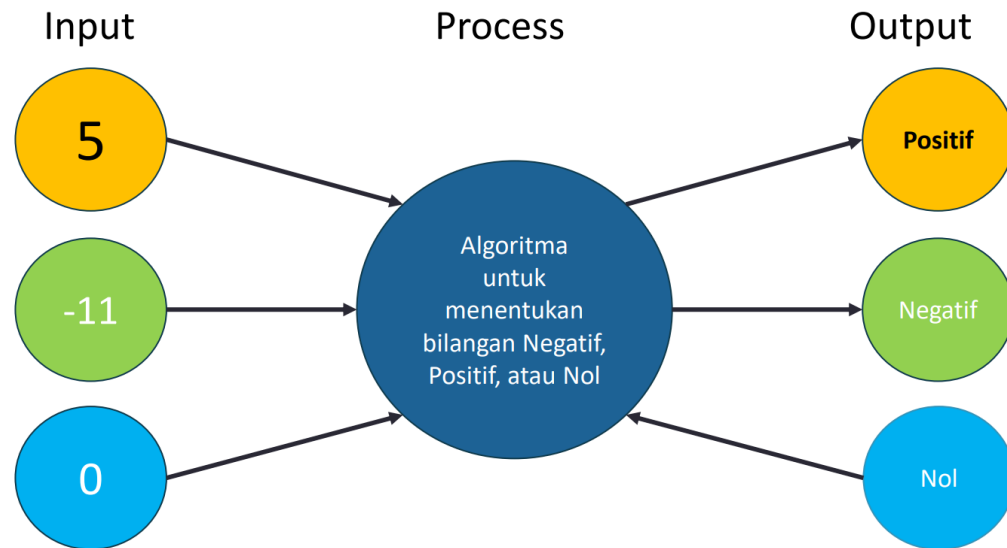
IPO (Input Process Output)

Konsep Dasar Input, Process, dan Output (IPO)

- Konsep input, process, dan output adalah prinsip dasar dalam pemrograman dan pengembangan algoritma.
- Setiap algoritma melibatkan tiga tahap utama: mengambil data masukan (input), melakukan operasi atau pengolahan data (process), dan menghasilkan hasil akhir (output).
- Konsep ini menggambarkan bagaimana algoritma beroperasi untuk memproses informasi.



Gambaran IPO (Menentukan Bilangan)



Notasi Algoritma Flowchart

1. Flowchart adalah representasi visual atau diagram alir yang digunakan untuk menggambarkan langkahlangkah dan urutan proses suatu algoritma atau program.
2. Flowchart menyajikan langkah-langkah dalam bentuk simbol-simbol grafis yang saling terhubung, membantu dalam memvisualisasikan bagaimana informasi mengalir dan bagaimana proses dilakukan.
3. Dalam kaitannya dengan notasi deskriptif, notasi algoritma yang menggunakan flowchart dapat lebih cepat dibaca dan dilihat alur dan hubungannya.

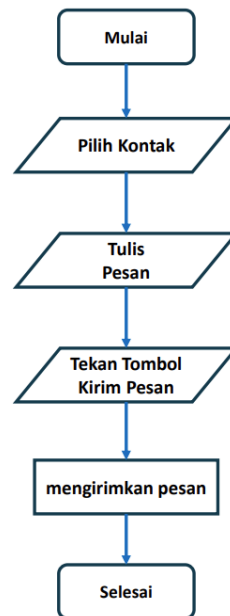
Simbol-simbol pada Flowchart

1. Setiap elemen flowchart dihubungkan oleh garis aliran bertanda panah
2. Garis aliran dimulai dari atas symbol dan keluar dari bagian bawah, kecuali symbol keputusan yang alirannya keluar dari bawah atau samping
3. Aliran bergerak dari atas ke bawah
4. Proses awal dan akhir menggunakan symbol terminal.



Contoh sederhana
Penggunaan *flowchart*
untuk menunjukan algoritma

Kasus/Aliran:
Mengirim pesan WhatsApp



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.

IPO (input process output)



Input

- List angka: [9, 8, 5, 3, 4, 7, 2, 1, 6]
- Target: 8

Process

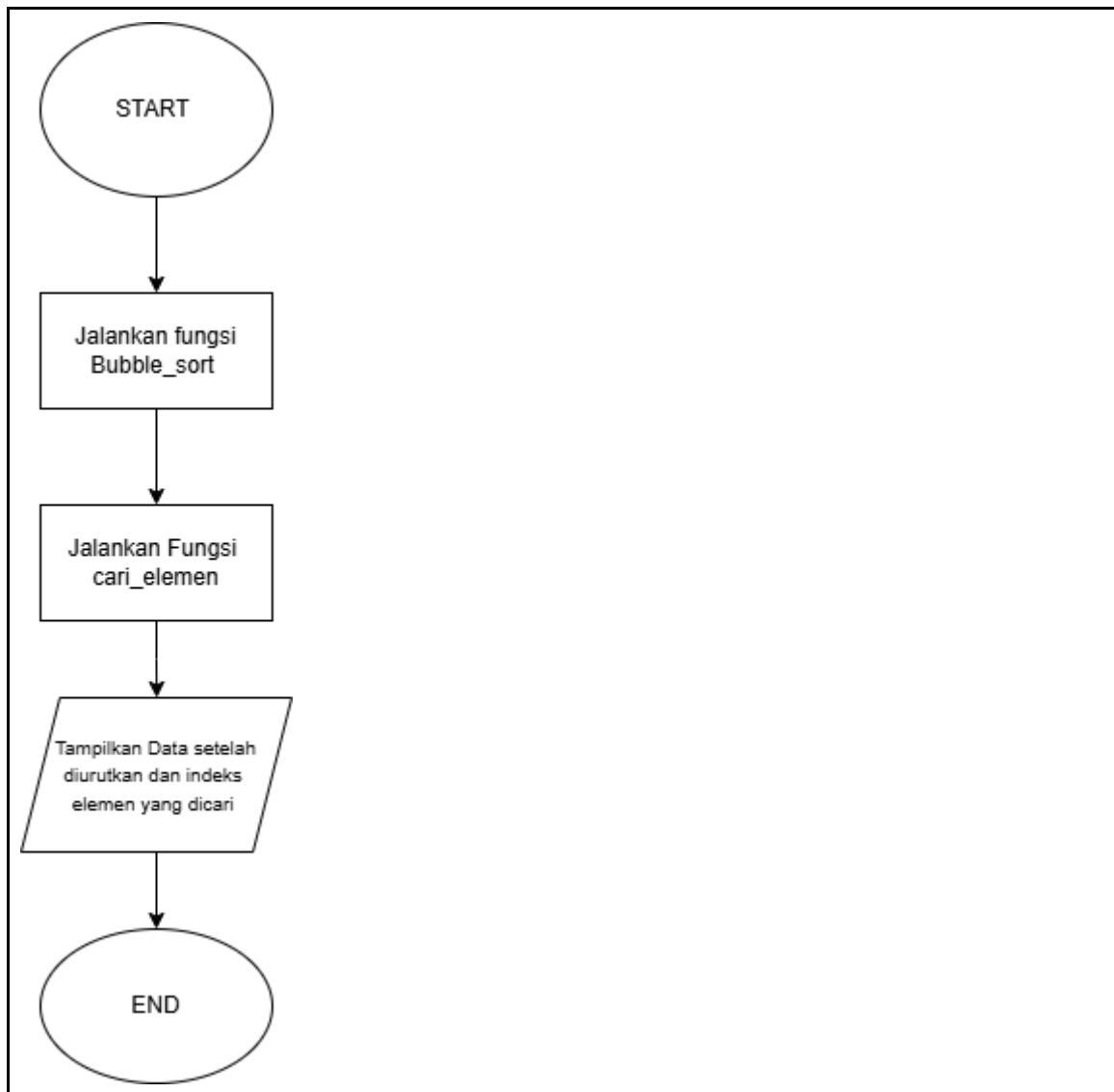
1. Mengurutkan list dengan Bubble Sort.
2. Mencari target menggunakan Binary Search.

Output

- List terurut: [1, 2, 3, 4, 5, 6, 7, 8, 9]
- Indeks target: 7

Flowchart





Source Code



```
def bubble_sort(list):
    n = len(list)
    for i in range(n):
        for j in range(0, n-i-1):
            if list[j] > list[j+1]:
                list[j], list[j+1] = list[j+1], list[j]

def cari_elemen(arr,target):
    bubble_sort(arr)
    print("Data Setelah Diurutkan:", arr)
    return binary_search(arr,target)

list1 = [9, 8, 5, 3, 4, 7, 2, 1, 6]
target = 8
print(f"Elemen {target} ditemukan di indeks {cari_elemen(list1, target)}.")
```

Output

```
Data Setelah Diurutkan: [1, 2, 3, 4, 5, 6, 7, 8, 9]
Elemen 8 ditemukan di indeks 7.
```

b. Latihan Kedua

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

IPO



Input

1. arr (list): Daftar bilangan yang ingin diurutkan.
2. n (integer, opsional): Panjang dari arr.

Process

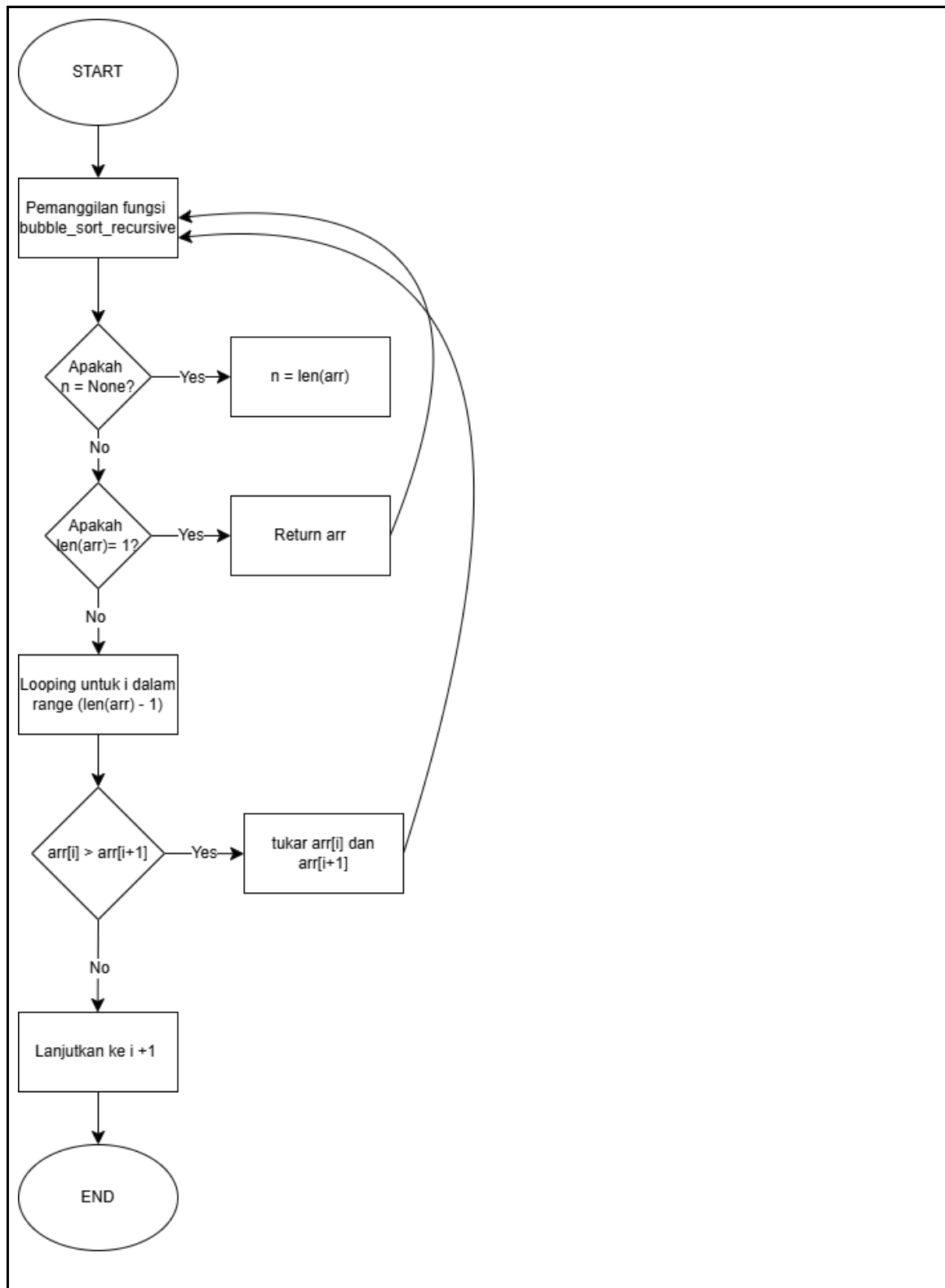
1. Jika n adalah None, set n ke panjang arr.
2. Jika panjang(arr) == 1, return arr.
3. Iterasi dari i dalam rentang dari 0 hingga n-1.
 - o Bandingkan arr[i] dan arr[i + 1].
 - o Jika arr[i] > arr[i + 1], tukar elemen tersebut.
4. Panggil kembali bubble_sort_recursive(arr, n - 1).

Output

1. Kembalikan arr yang sudah diurutkan.
2. Cetak "Hasil pengurutan:", diikuti oleh arr yang telah diurutkan.

Flowchart





Source Code

```

def bubble_sort_recursive(arr, n=None):
    if n is None:
        n = len(arr)
    if n == 1:
        return arr
    for i in range(n - 1):
        if arr[i] > arr[i + 1]:
            arr[i], arr[i + 1] = arr[i + 1], arr[i]
    return bubble_sort_recursive(arr, n - 1)

data = [45, 2, 26, 59, 16, 77, 125]
sorted_data = bubble_sort_recursive(data)
print("Hasil pengurutan:", sorted_data)

```

Output

```

Hasil pengurutan: [2, 16, 26, 45, 59, 77, 125]

```



4. File Praktikum

Github Repository:

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. Algoritma binary search hanya dapat bekerja dengan baik pada data yang sudah diurutkan. Ini karena binary search menggunakan prinsip pembagian setengah untuk mencari, yang hanya efektif jika data sudah tersusun. Jika data tidak diurutkan, binary search tidak bisa bekerja karena akan menghasilkan hasil pencarian yang tidak akurat.
2. Fungsi bubble_sort_recursive menerima sebuah list dan mengurutkannya menggunakan metode bubble sort secara rekursif. Pertama, jika panjang list (n) tidak diberikan, panjang list dihitung. Jika panjang list adalah 1, maka list sudah terurut dan dikembalikan. Proses sorting dilakukan dengan membandingkan setiap pasangan elemen dan menukar elemen yang tidak sesuai urutan. Setelah itu, fungsi dipanggil kembali dengan n-1 untuk melanjutkan sorting hingga seluruh list terurut. Hasil akhir adalah list yang terurut, yang kemudian dicetak sebagai "Hasil pengurutan:".

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.
- b. Kita dapat mengetahui bahwa dalam algoritma pencarian, penting untuk memiliki data yang terurut agar metode binary search dapat bekerja dengan efektif. Selain itu, penggunaan algoritma sorting seperti bubble sort secara rekursif dalam bahasa pemrograman Python membantu mempermudah proses pengolahan data. Ketelitian dalam memasukkan fungsi dan memahami alur kerja algoritma sangat penting untuk menghasilkan keluaran yang tepat dan akurat.



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	20 Menit	Menarik
2.	Latihan Kedua	20 Menit	Menarik

Keterangan:

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