# LAPORAN PRAKTIKUM ALGORITMA PEMROGRAMAN 2



# NIM 2200016103

# NAMA : ABDI SETIAWAN

# RUANG : D

# PERTEMUAN 6

**PROGRAM STUDI SISTEM INFORMASI FAKULTAS SAINS DAN TEKNOLOGI TERAPAN UNIVERSITAS AHMAD DAHLAN YOGYAKARTA**

# TAHUN AJARAN 2022/2023

## Input Process Output (IPO) dan Kode Pemrograman.

1. **Sequential Search**

**def** sqesearch(**data**,**key**): pos **=** []

hitung **=** 0

**for** i **in** range(len(data)): hitung **+=** 1

**if** data[i] **==** key: pos.append(i**+**1)

**if** len(pos)**>**0:

**print**("Data", key, "Sebanyak", len(pos), "ditemukan di posisi", pos)

**else**:

**print**("Data tidak ditemukan")

**print**("Jumlah iterasi yang diperlukan:", hitung)

**return** pos

Input:

* + **data**: Sebuah list yang berisi data untuk pencarian.
  + **key**: Data yang akan dicari di dalam **data**. Proses:

1. Menginisialisasi list **pos** sebagai list kosong untuk menyimpan posisi kemunculan

**key**.

1. Menginisialisasi variabel **hitung** dengan nilai 0 untuk melacak jumlah iterasi.
2. Melakukan iterasi sebanyak **len(data)** kali menggunakan loop **for**.
3. Di setiap iterasi, menambahkan 1 ke variabel **hitung**.
4. Jika data pada indeks ke-i dalam **data** == **key**, maka:
   * Menambahkan posisi (**i+1**) ke dalam list **pos**.
5. Jika panjang list **pos** lebih dari 0, maka:
   * Mencetak pesan yang menyatakan bahwa data **key** ditemukan sebanyak

**len(pos)** di posisi-posisi **pos**.

1. Jika panjang list **pos** sama dengan 0, maka:
   * Mencetak pesan yang menyatakan bahwa data **key** tidak ditemukan.
2. Mencetak jumlah iterasi yang diperlukan.
3. Mengembalikan list **pos** yang berisi posisi-posisi kemunculan **key** dalam **data**. Output:
   * Jika data **key** ditemukan di dalam **data**, maka outputnya adalah pesan yang menyatakan data **key** ditemukan sebanyak **len(pos)** kali di posisi-posisi **pos**.
   * Jika data **key** tidak ditemukan di dalam **data**, maka outputnya adalah pesan yang menyatakan bahwa data **key** tidak ditemukan.
   * Jumlah iterasi yang diperlukan juga dicetak sebagai output.
   * Nilai yang dikembalikan adalah list **pos** yang berisi posisi-posisi kemunculan **key**

dalam **data**.

## Binary Search

**def** binsearch(**data**,**key**): awal **=** 1

akhir **=** len(data)**+**1 ketemu **=** False hitung **=** 0

**while** (awal **<=** akhir) **and not** ketemu: hitung **+=** 1

tengah **=** int((awal**+**akhir)**/**2)

**if** key **==** data[tengah]: ketemu **=** True

**print**("data", key,"ditemukan di indeks ke", tengah)

**elif** (key**<**data[tengah]): akhir **=** tengah **-** 1

**else**:

awal **=** tengah **+** 1

**if** ketemu **==** False:

**print**("data tidak ditemukan")

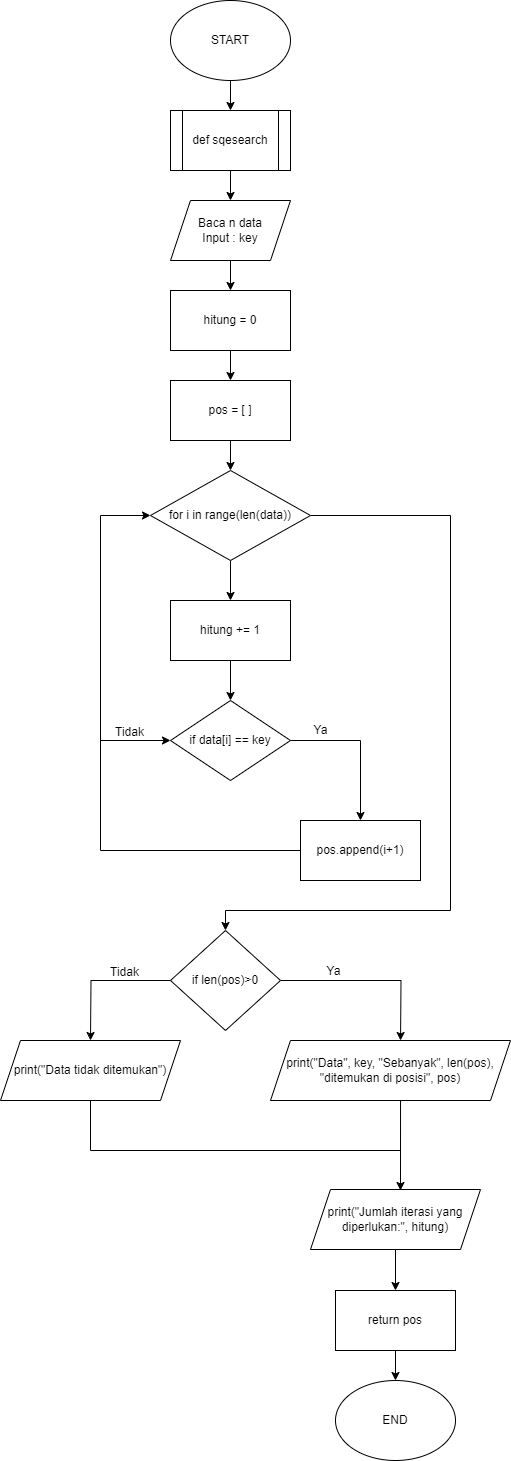
**print**("Jumlah iterasi yang diperlukan:", hitung)

Input:

* + **data**: Daftar elemen yang telah diurutkan.
  + **key**: Elemen yang akan dicari dalam daftar **data**. Proses:
  + Inisialisasikan **awal** dengan nilai 1, yang menunjukkan indeks awal dari daftar.
  + Inisialisasikan **akhir** dengan nilai **len(data) + 1**, yang menunjukkan indeks akhir dari daftar.
  + Inisialisasikan **ketemu** sebagai False, yang menunjukkan apakah **key** ditemukan atau tidak.
  + Inisialisasikan **hitung** dengan nilai 0, yang merepresentasikan jumlah iterasi.
  + Selama **awal** kurang dari atau sama dengan **akhir** dan **ketemu** adalah False:
    - Tambahkan **hitung** dengan 1.
    - Hitung indeks **tengah** sebagai titik tengah antara **awal** dan **akhir**.
    - Jika **key** sama dengan **data[tengah]**:
      * Set **ketemu** menjadi True.
      * Cetak bahwa **key** ditemukan pada indeks **tengah**.
    - Jika **key** lebih kecil dari **data[tengah]**, perbarui **akhir** menjadi **tengah - 1**.
    - Jika **key** lebih besar dari **data[tengah]**, perbarui **awal** menjadi **tengah + 1**.
  + Jika **ketemu** masih False, cetak bahwa **key** tidak ditemukan.
  + Cetak jumlah total iterasi (**hitung**). Output:
  + Jika **key** ditemukan dalam daftar **data**, akan mencetak indeks di mana **key** ditemukan.
  + Jika **key** tidak ditemukan, akan mencetak bahwa data tidak ditemukan.
  + Juga akan mencetak jumlah total iterasi yang diperlukan untuk melakukan pencarian.

## Flowchart

* 1. **Sequential Search**



## Binary Search

