GuruVirtual.ID

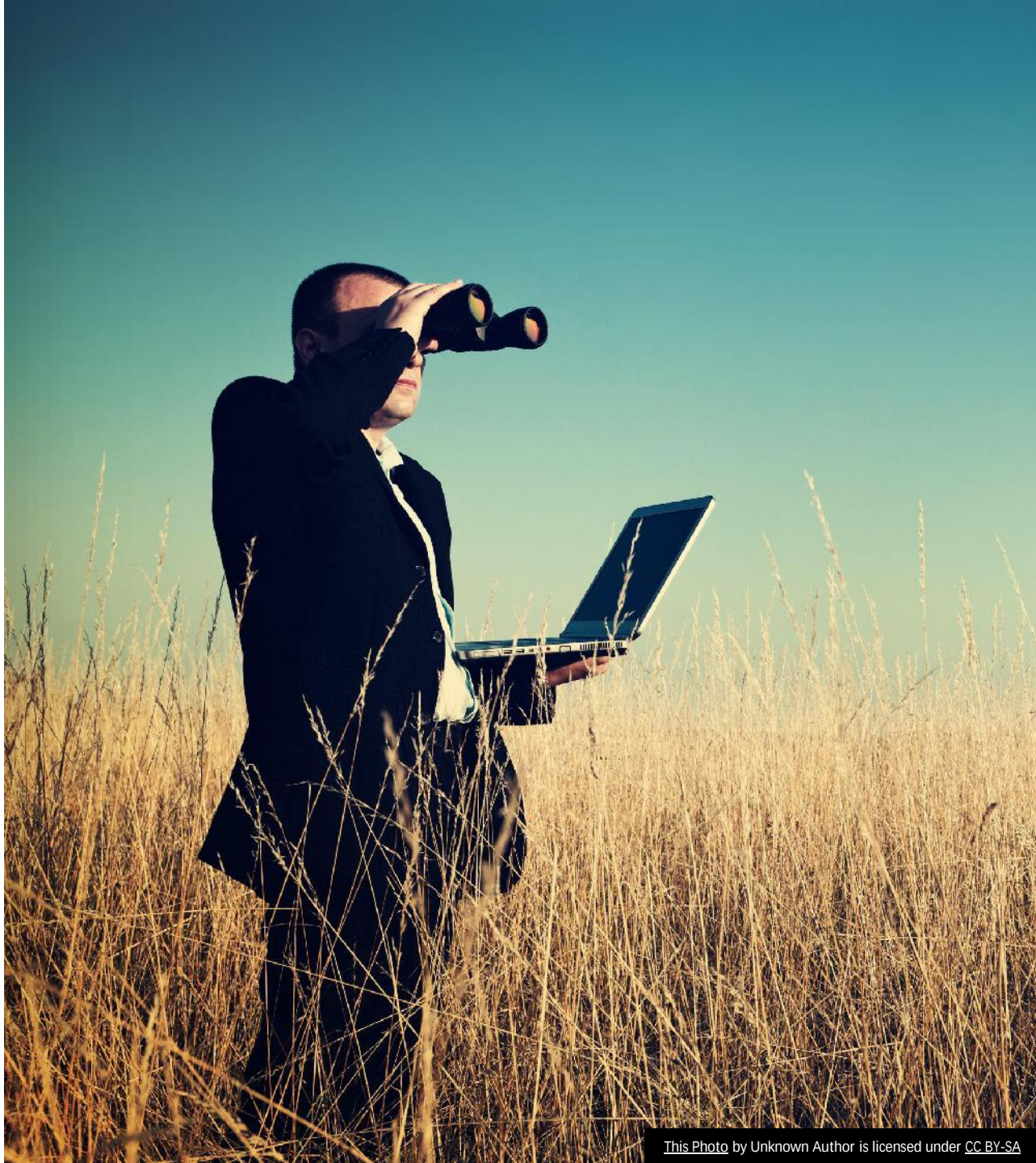
Implementasi Algoritma Pencarian Linear, Binary, dan Hash Menggunakan Python

Hartono, S.Pd., M.T.I

Universitas Muhammadiyah Kotabumi

Algoritma Pencarian

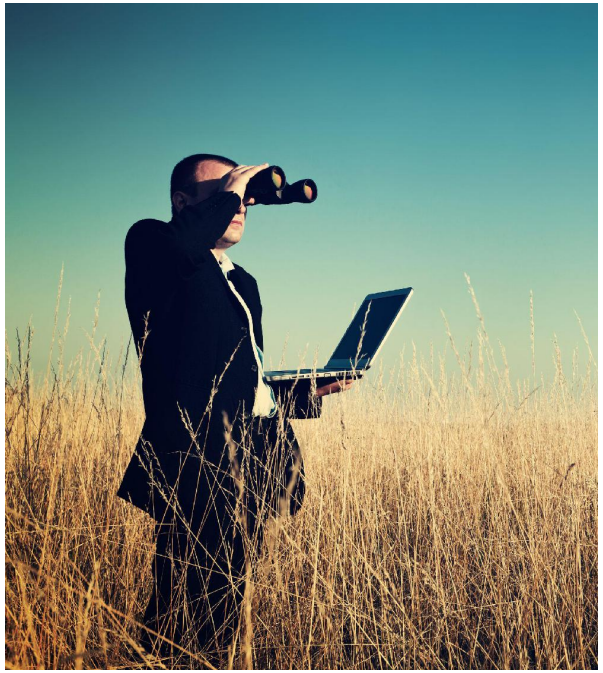
- Dalam pemrograman, algoritma pencarian digunakan untuk menemukan lokasi atau keberadaan suatu nilai atau elemen dalam kumpulan data tertentu.
- Pencarian dapat membantu mengambil informasi yang diinginkan dari data yang besar atau kompleks.
- Ada beberapa jenis algoritma pencarian yang digunakan tergantung pada karakteristik data dan tujuan pencarian.



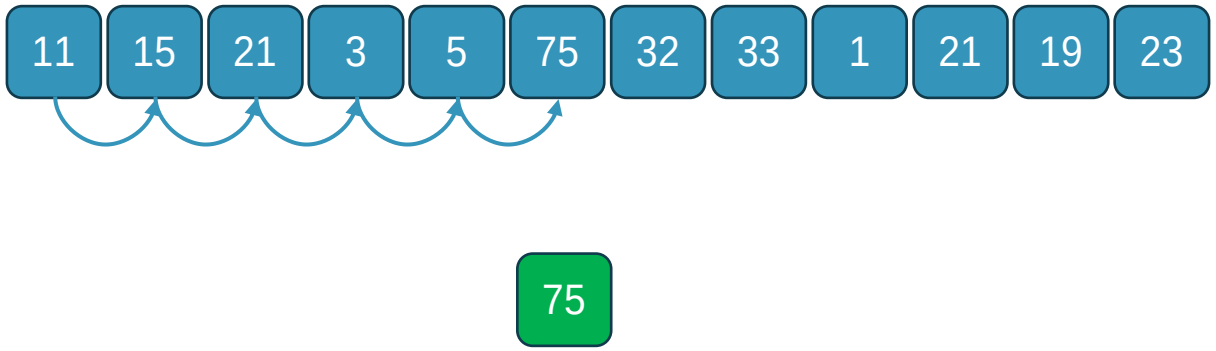
Jenis-Jenis Algoritma Pencarian

1. Pencarian Linear (Sequential Search)
2. Pencarian Binary
3. Pencarian Hashing

1. Pencarian Linear (Sequential Search)



- Pencarian Linear, juga dikenal sebagai Sequential Search, adalah algoritma pencarian sederhana di mana setiap elemen dalam dataset diperiksa secara berurutan sampai nilai yang dicari ditemukan atau seluruh dataset telah diperiksa.
- Ini adalah salah satu metode pencarian yang paling sederhana dan mudah dipahami, namun mungkin tidak efisien untuk dataset yang besar.



Cara Kerja Linear/Sequential Search

- Dimulai dari elemen pertama dalam dataset.
- Setiap elemen diperiksa untuk kesesuaian dengan nilai yang dicari.
- Jika elemen yang sesuai ditemukan, pencarian berhenti dan posisi elemen ditemukan dikembalikan.
- Jika seluruh dataset diperiksa dan elemen tidak ditemukan, pencarian mengembalikan indikasi bahwa nilai tidak ditemukan.

Kelebihan Pencarian Linear

- Mudah dipahami dan diimplementasikan.
- Cocok untuk dataset yang relatif kecil.
- Bekerja baik untuk data yang tidak terurut.

Kekurangan Pencarian Linear

- Memerlukan waktu yang proporsional dengan ukuran dataset.
- Tidak efisien untuk dataset besar karena memerlukan banyak operasi perbandingan.

```python
def linear_search(arr, target):
    for i in range(len(arr)):
        if arr[i] == target:
            return i  # Mengembalikan indeks elemen yang
    return -1  # Mengembalikan -1 jika nilai tidak ditemu

data = [5, 8, 12, 6, 2, 15]
target_value = 6

result = linear_search(data, target_value)

if result != -1:
    print(f"Nilai {target_value} ditemukan pada indeks {
else:
    print(f"Nilai {target_value} tidak ditemukan dalam d
```

Linear Search Python Coding

Guru Virtual

Pencarian Binary

- Pencarian Binary adalah algoritma pencarian yang efisien untuk dataset yang sudah diurutkan.
- Algoritma ini bekerja dengan membagi dataset menjadi dua bagian dan memeriksa elemen tengah.
- Jika elemen tengah tidak cocok dengan nilai yang dicari, algoritma akan memutuskan bagian mana dari dataset yang harus diabaikan berdasarkan hubungan antara elemen tengah dan nilai yang dicari.
- Langkah ini diulang sampai elemen yang sesuai ditemukan atau seluruh dataset diperiksa.

Mencari Angka 70 (Pencarian Biner)

Cara Kerja

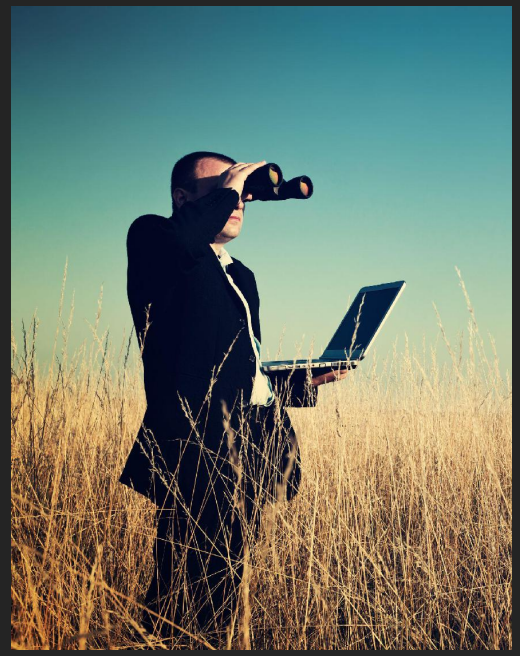
- Tentukan elemen tengah dataset.
- Periksa apakah elemen tengah cocok dengan nilai yang dicari.
- Jika cocok, pencarian berhenti dan posisi elemen ditemukan dikembalikan.
- Jika nilai elemen tengah lebih besar dari nilai yang dicari, cari di setengah kiri dataset.
- Jika nilai elemen tengah lebih kecil dari nilai yang dicari, cari di setengah kanan dataset.
- Ulangi langkah-langkah di atas sampai nilai ditemukan atau seluruh dataset diperiksa.

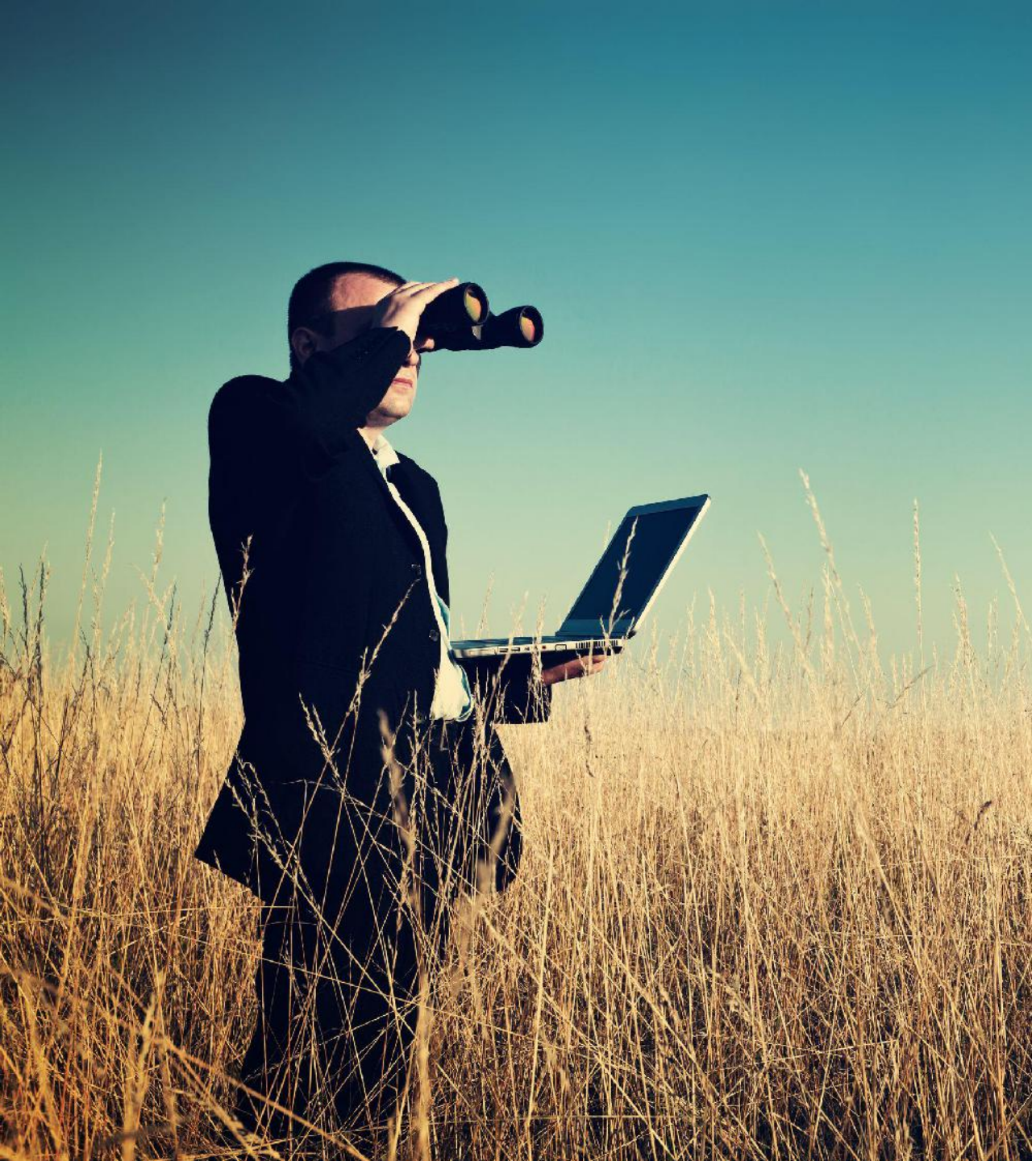
Binary Search Python Coding

```python
def binary_search(arr, target):
    low = 0
    high = len(arr) - 1

    while low <= high:
        mid = (low + high) // 2
        if arr[mid] == target:
            return mid  # Mengembalikan indeks elemen yang sesuai
        elif arr[mid] < target:
            low = mid + 1
        else:
            high = mid - 1
    return -1  # Mengembalikan -1 jika nilai tidak ditemukan

data = [2, 5, 6, 8, 12, 15]
target_value = 8

result = binary_search(data, target_value)

if result != -1:
    print(f"Nilai {target_value} ditemukan pada indeks {result}.")
else:
    print(f"Nilai {target_value} tidak ditemukan dalam dataset.")
```

Kelebihan Pencarian Binary

- Efisien untuk dataset besar karena mengurangi jumlah operasi perbandingan.
- Hanya berfungsi pada data yang diurutkan.
- Waktu eksekusi lebih cepat dibandingkan dengan pencarian linear untuk dataset besar.





Kekurangan Pencarian Binary

- Hanya cocok untuk dataset yang sudah diurutkan.
- Implementasi lebih kompleks daripada pencarian linear.
- Tidak efektif jika data tidak diurutkan.

Pencarian Hashing

- Pencarian Hashing adalah algoritma pencarian yang menggunakan struktur data yang disebut tabel hash (hash table) untuk menyimpan dan mengakses data dengan efisien.
- Algoritma ini memanfaatkan fungsi hash untuk mengubah nilai pencarian menjadi indeks dalam tabel hash, di mana elemen yang sesuai diharapkan berada.

Cara Kerja

- Data dimasukkan ke dalam tabel hash menggunakan fungsi hash. Fungsi ini mengonversi nilai pencarian menjadi indeks dalam tabel.
- Saat pencarian dilakukan, nilai pencarian diubah menjadi indeks melalui fungsi hash.
- Algoritma mengunjungi indeks yang dihasilkan untuk mencari nilai yang sesuai.
- Jika nilai ditemukan pada indeks tersebut, pencarian berhasil.
- Jika terjadi tabrakan hash (dua nilai dihasilkan oleh fungsi hash yang sama), algoritma harus memiliki strategi untuk mengatasi konflik tersebut.

Kekurangan

- Memerlukan manajemen tabrakan hash.
- Fungsi hash yang buruk dapat mengurangi efisiensi.
- Tidak cocok untuk data yang sering berubah.

Kekurangan

- Memerlukan manajemen tabrakan hash.
- Fungsi hash yang buruk dapat mengurangi efisiensi.
- Tidak cocok untuk data yang sering berubah.

Kenapa Pencarian Berbasis Hash Cepat?

- Fungsi hash diimplementasikan untuk membuat record yang besar memiliki “kode” yang lebih kecil ke dalam indeks, sehingga mempercepat pencarian.
- Tujuan melakukan hashing agar tiap record memiliki nilai hash yang sama.

Gambaran Pencarian Hash

Hal-hal yang Tentang Pencarian

- Efisiensi Pencarian: Algoritma pencarian membantu mengoptimalkan waktu dan sumber daya yang diperlukan untuk menemukan nilai dalam dataset besar.
- Pengindeksan dan Basis Data: Dalam basis data, algoritma pencarian digunakan untuk mengambil data dari tabel atau indeks, mempercepat pencarian data tertentu.
- Pengelolaan Inventori: Dalam bisnis, pencarian digunakan untuk melacak persediaan, mencari produk, atau informasi tertentu dalam gudang atau toko.
- Pencarian dalam Struktur Data: Algoritma pencarian sering digunakan dalam struktur data seperti array, linked list, dan pohon untuk menemukan elemen tertentu.
- Optimisasi Algoritma Lain: Beberapa algoritma lain seperti pengurutan menggunakan algoritma pencarian untuk memeriksa keberadaan elemen tertentu dalam proses pengurutan.
- Pencarian Informasi: Dalam basis data besar atau di web, algoritma pencarian digunakan untuk menemukan informasi yang relevan dari data yang luas.