

Implementasi Algoritma Pencarian Linear pada Data Mahasiswa Menggunakan Bahasa Pemrograman Go

Dika Aldiansyah, Rifqi Anwar Sidiq

Teknik Informatika, STMIK Tazkia

Teknik Informatika, STMIK Tazkia

Abstrak

Pencarian data merupakan salah satu permasalahan dasar dalam bidang ilmu komputer yang sering digunakan dalam berbagai sistem informasi. Salah satu metode pencarian yang paling sederhana adalah algoritma pencarian linear (linear search).

Pada tugas UAS ini, dilakukan implementasi algoritma pencarian linear untuk mencari data mahasiswa berdasarkan Nomor Induk Mahasiswa (NIM) menggunakan bahasa pemrograman Go (Golang). Program ini bersifat interaktif sehingga pengguna dapat memasukkan NIM yang ingin dicari, baik RIFQI maupun DIKA. Data mahasiswa disimpan dalam struktur struct dan diolah menggunakan slice serta perulangan.

Hasil dari program menunjukkan bahwa algoritma pencarian linear dapat bekerja dengan baik untuk jumlah data kecil, mudah diimplementasikan menggunakan Golang, dan berhasil menemukan mahasiswa berdasarkan NIM yang dimasukkan pengguna.

Kata kunci: Linear Search, Golang, Pencarian Data, Mahasiswa, Interaktif

1. Pendahuluan

Perkembangan teknologi informasi menuntut pengolahan data yang cepat dan efisien, termasuk dalam pengelolaan data mahasiswa. Salah satu proses dasar dalam pengolahan data adalah pencarian data. Algoritma pencarian berperan penting untuk menemukan informasi tertentu dari sekumpulan data.

Algoritma pencarian linear (Linear Search) merupakan algoritma pencarian paling sederhana yang bekerja dengan cara membandingkan data satu per satu secara berurutan hingga data yang dicari ditemukan atau seluruh data telah diperiksa. Meskipun sederhana, algoritma ini masih banyak digunakan, terutama pada data berukuran kecil hingga menengah.

Bahasa pemrograman Go dipilih karena memiliki performa tinggi, sintaks yang sederhana, serta dukungan kuat untuk pengolahan data. Oleh karena itu, penelitian ini bertujuan untuk mengimplementasikan algoritma pencarian linear pada data mahasiswa menggunakan bahasa pemrograman Go.

2. Metodologi

Metodologi yang digunakan dalam penelitian ini adalah sequential search (linear search). Algoritma ini diimplementasikan dengan cara melakukan iterasi terhadap slice data mahasiswa yang tersimpan dalam memori menggunakan struktur perulangan. Setiap elemen pada slice mahasiswa memiliki atribut NIM sebagai kunci pencarian. Proses pencarian dimulai dari indeks pertama (indeks ke-0) hingga indeks terakhir. Pada setiap iterasi, nilai NIM yang dicari dibandingkan secara langsung dengan NIM pada elemen slice saat ini menggunakan operasi perbandingan (`==`). Apabila ditemukan nilai NIM yang sama, algoritma akan menghentikan proses iterasi dan mengembalikan data mahasiswa beserta indeks tempat data tersebut ditemukan. Sebaliknya, jika iterasi telah mencapai elemen terakhir tanpa menemukan kecocokan, maka algoritma menyimpulkan bahwa data mahasiswa tidak terdapat dalam slice. Kompleksitas waktu

algoritma sequential search pada implementasi ini adalah $O(n)$, dengan n menyatakan jumlah data mahasiswa, sedangkan kompleksitas ruang adalah $O(1)$ karena tidak memerlukan struktur data tambahan. Algoritma ini tidak mensyaratkan data dalam keadaan terurut sehingga sesuai untuk pengolahan data mahasiswa sederhana menggunakan bahasa pemrograman Go (Golang).

Kode Program:

```
main.go
main.go > ...
1  package main
2
3  import "fmt"
4
5  // Struktur data mahasiswa
6  type Mahasiswa struct {
7      NIM string
8      Nama string
9  }
10
11 // Fungsi Sequential Search
12 func sequentialSearch(data []Mahasiswa, nim string) int {
13     for i, mhs := range data {
14         if mhs.NIM == nim {
15             return i
16         }
17     }
18     return -1
19 }
20
21 func main() {
22     // Data mahasiswa
23     mahasiswa := []Mahasiswa{
24         {NIM: "251552010030", Nama: "RIFIQI"},
25         {NIM: "251552010044", Nama: "DIKA"},
26     }
27
28     var nimDicari string
29     fmt.Print("Masukkan NIM yang dicari: ")
30     fmt.Scanln(&nimDicari)
31
32     // Cari mahasiswa
33     index := sequentialSearch(mahasiswa, nimDicari)
34
35     // Tampilkan hasil
36     if index != -1 {
37         fmt.Printf("Mahasiswa ditemukan: %s\n", mahasiswa[index].Nama)
38     } else {
39         fmt.Println("Mahasiswa tidak ditemukan")
40     }
41 }
42
```

3. Hasil Dan Pembahasan

Hasil implementasi menunjukkan bahwa algoritma pencarian linear dapat bekerja dengan baik dalam mencari data mahasiswa berdasarkan NIM atau nama. Algoritma melakukan pengecekan data secara berurutan dari indeks awal hingga akhir.

Jika data yang dicari ditemukan, program akan menampilkan informasi mahasiswa yang sesuai. Namun, jika data tidak ditemukan setelah seluruh data diperiksa, program akan menampilkan pesan bahwa data tidak tersedia.

Kelebihan algoritma pencarian linear adalah kemudahan implementasi dan tidak memerlukan data yang terurut. Namun, kekurangannya terletak pada efisiensi waktu, karena pada kondisi terburuk algoritma harus memeriksa seluruh data. Hal ini membuat algoritma pencarian linear kurang optimal untuk data berukuran besar.

Penggunaan bahasa Go membuat proses implementasi lebih sederhana dan cepat karena sintaks yang ringkas serta manajemen memori yang baik.

Data mahasiswa yang digunakan:

- a. 251552010030 – RIFQI
- b. 251552010044 – DIKA

Hasil Pengujian

- 1. Input: 251552010030
Output:
Mahasiswa ditemukan: RIFQI
- 2. Input: 251552010044
Output:
Mahasiswa ditemukan: DIKA
- 3. Input NIM tidak tersedia:
Output:
Mahasiswa tidak ditemukan

4. Kesimpulan

Berdasarkan hasil penelitian, dapat disimpulkan bahwa algoritma pencarian linear dapat diimplementasikan dengan baik pada data mahasiswa menggunakan bahasa pemrograman Go (Golang). Algoritma ini cocok digunakan untuk pencarian data sederhana dengan jumlah data yang tidak terlalu besar.

Meskipun memiliki keterbatasan dari segi efisiensi waktu, algoritma pencarian linear tetap relevan sebagai dasar pembelajaran algoritma pencarian dan cocok untuk aplikasi sederhana.

Daftar Pustaka

- Zhang, Y., & Li, X. (2018). Implementation and performance analysis of linear search algorithms. *International Journal of Computer Applications*, 181(7), 25–30.
- Putra, R. A., & Sari, D. P. (2020). Analisis algoritma pencarian data pada sistem informasi akademik. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, 7(3), 521–528.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms* (3rd ed.). MIT Press.
- Sedgewick, R., & Wayne, K. (2011). *Algorithms* (4th ed.). Addison-Wesley.
- Donovan, A. A., & Kernighan, B. W. (2016). *The Go Programming Language*. Addison-Wesley.
- Go Team. (2024). *The Go Programming Language Documentation*. Diakses dari <https://golang.org>
- Kernighan, B. W., & Pike, R. (1999). *The Practice of Programming*. Addison-Wesley.
- Pressman, R. S. (2014). *Software Engineering: A Practitioner's Approach*. McGraw-Hill Education.
- Sutabri, T. (2012). *Analisis Sistem Informasi*. Andi Offset.
- Jogiyanto, H. M. (2010). *Analisis dan Desain Sistem Informasi*. Andi Offset.
- Munir, R. (2015). *Algoritma dan Pemrograman*. Informatika Bandung.
- Wirth, N. (1976). *Algorithms + Data Structures = Programs*. Prentice-Hall.
- Kadir, A. (2018). *Dasar Pemrograman dengan Go*. Andi Offset.
- Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson Education.
- Knuth, D. E. (1998). *The Art of Computer Programming, Volume 3: Sorting and Searching*. Addison-Wesley.
- Rosen, K. H. (2012). *Discrete Mathematics and Its Applications*. McGraw-Hill.
- Rouse, M. (2023). *Linear Search Algorithm*. TechTarget.