

# ANALISA ALGORITMA PADA METODA Pencarian LINIER, BINER DAN INTERPOLASI

Harold Situmorang

Program Studi Sistem Informasi Universitas Sari Mutiara Indonesia

[Haroldsitumorang@gmail.com](mailto:Haroldsitumorang@gmail.com)

## ABSTRAK

Dalam ilmu komputer terdapat bermacam – macam algoritma untuk metoda pencarian (*searching*). Beberapa metoda pencarian yang pernah dipelajari adalah metoda pencarian linier (*Linear / Sequential Search*), pencarian biner (*Binary Search*) dan pencarian interpolasi (*Interpolation Search*). Masing – masing algoritma memiliki prasyarat dan cara serta waktu pelaksanaan yang berbeda. Pemilihan atas metoda pencarian dilakukan berdasarkan keadaan dan keinginan pengguna metoda yang biasanya tergantung pada jumlah data, jenis data dan struktur data yang digunakan. Pencarian (*searching*) merupakan suatu pekerjaan yang sering dikerjakan dalam kehidupan sehari – hari. Ada kalanya kita mencari sesuatu dengan tujuan hanya untuk mengetahui apakah data tersebut ada dalam sekumpulan data atau tidak, sementara di lain waktu mungkin kita menginginkan posisi dari data yang dicari tersebut. Pada tahap simulasi dan analisa, terdapat prosedur ‘SimulasiCari’, yang berfungsi untuk mensimulasikan tahapan – tahapan pencarian dan menampilkan hasil analisis dan prosedur kerja program dalam bentuk *report*. Prosedur ‘SimulasiCari’ terdiri atas 3 prosedur utama, yaitu ‘*LinearSearch*’, ‘*BinarySearch*’ dan ‘*InterpolationSearch*’. Prosedur ‘*LinearSearch*’ berfungsi untuk mensimulasikan tahapan pencarian dengan algoritma pencarian linier dan menampilkan hasil analisis dan prosedur kerja program dalam bentuk *report*. Prosedur ‘*BinarySearch*’ berfungsi untuk mensimulasikan tahapan pencarian dengan algoritma pencarian biner dan menampilkan hasil analisis dan prosedur kerja program dalam bentuk *report*. Prosedur ‘*InterpolationSearch*’ berfungsi untuk mensimulasikan tahapan pencarian dengan algoritma pencarian interpolasi dan menampilkan hasil analisis dan prosedur kerja program dalam bentuk *report*.

**Kata Kunci** : Pencarian Linier (*Linear / Sequential Search*), Pencarian Biner (*Binary Search*)  
Dan Pencarian Interpolasi (*Interpolation Search*)

## I. PENDAHULUAN

### 1.1 Latar Belakang

Pencarian (*searching*) merupakan suatu pekerjaan yang sering dikerjakan dalam kehidupan sehari – hari. Ada kalanya kita mencari sesuatu dengan tujuan hanya untuk mengetahui apakah data tersebut ada dalam sekumpulan data atau tidak, sementara di lain waktu mungkin kita menginginkan posisi dari data yang dicari tersebut.

Dalam ilmu komputer terdapat bermacam – macam algoritma untuk metoda pencarian (*searching*). Beberapa metoda pencarian yang pernah dipelajari adalah metoda pencarian linier (*Linear / Sequential Search*), pencarian biner (*Binary Search*) dan pencarian interpolasi (*Interpolation Search*). Masing – masing algoritma memiliki prasyarat dan cara serta waktu pelaksanaan yang berbeda. Pemilihan atas metoda pencarian dilakukan berdasarkan keadaan dan

keinginan pengguna metoda yang biasanya tergantung pada jumlah data, jenis data dan struktur data yang digunakan.

Berdasarkan uraian di atas, penulis bermaksud untuk merancang suatu perangkat lunak yang mampu untuk melakukan analisa terhadap algoritma dari metoda pencarian linier, biner dan interpolasi dari data – data numerik yang di-*input*.

### 1.2 Rumusan Masalah

Berdasarkan latar belakang pemilihan judul, maka yang menjadi permasalahan adalah :

1. Menampilkan cara kerja dari algoritma metoda pencarian linier, biner dan interpolasi secara tahap demi tahap.
2. Membuat suatu perangkat lunak untuk menganalisa algoritma dari metoda pencarian linier, biner dan interpolasi dengan menggunakan bahasa pemrograman *Visual Basic 6.0*.

### 1.3 Batasan Masalah

Karena keterbatasan waktu dan pengetahuan penulis, maka ruang lingkup permasalahan dalam merancang perangkat lunak ini antara lain :

1. *Input* dibatasi hanya pada *keyboard* sedangkan *Output* perangkat lunak pada layar monitor.
2. Data yang di-*input* berupa numerik *integer*.
3. Jumlah data dibatasi maksimal 100 buah.
4. Data yang di-*input* akan diurutkan (menaik atau menurun) secara otomatis oleh

komputer tanpa menunjukkan tahap – tahap pengurutan.

5. Perangkat lunak dapat menampilkan tahapan pencarian secara tahap demi tahap.
6. Perangkat lunak juga memiliki fasilitas untuk menyimpan dan membuka kembali data – data yang di-*input*.

### 1.4 Tujuan Penulisan

Tujuan penelitian ini adalah untuk merancang suatu perangkat lunak yang mampu untuk melakukan analisa terhadap algoritma dari metoda pencarian linier, biner dan interpolasi untuk data terurut menaik (*ascending*) dan terurut menurun (*descending*) secara tahap demi tahap.

### 1.5 Manfaat Penelitian

Manfaat dari penelitian ini yaitu :

1. Membantu pembelajaran algoritma dari metoda pencarian linier, biner dan interpolasi.
2. Sebagai fasilitas pendukung dalam proses belajar – mengajar.

## II. LANDASAN TEORI

### 2.1 Pengenalan Algoritma

Kata algoritma berasal dari nama seorang ahli matematika berkebangsaan Persia yang hidup pada abad ke – 9 yang bernama Abu Abdullah Muhammad bin Musa Al-Khawarizmi. Pada awalnya, kata ‘algorism’ diartikan sebagai aturan – aturan untuk melakukan proses aritmatika menggunakan numerik Arab. Kata ‘algorism’ diubah menjadi kata ‘algorithm’ pada abad ke –

18. Sekarang, pengertian dari kata ini mencakup semua prosedur terhingga untuk menyelesaikan problema atau melakukan pekerjaan.

Penerapan pertama dari algoritma yang ditulis untuk sebuah komputer adalah ‘*Ada Byron’s notes on the analytical engine*’ yang ditulis pada tahun 1842, dimana Ada Byron dianggap oleh kebanyakan orang sebagai *programmer* pertama di dunia. Walaupun, sejak Charles Babbage tidak menyelesaikan *analytical engine*-nya, algoritma ini tidak pernah diimplementasikan lagi.

## 2.2 Pencarian

Pencarian (*searching*) merupakan suatu pekerjaan yang sering dikerjakan dalam kehidupan sehari – hari. Ada kalanya pencarian dilakukan dengan tujuan hanya untuk mengetahui apakah data tersebut ada dalam sekumpulan data atau tidak, atau mungkin di lain waktu posisi dari data yang dicari tersebut dibutuhkan untuk keperluan tertentu, atau jika kemunculan data lebih dari satu kali maka semua posisi dan frekuensi kemunculannya ingin ditampilkan.

Pencarian paling sederhana dapat digambarkan sebagai berikut,

Misalkan suatu barisan data  $A[1] \dots A[n]$ , maka yang menjadi perproblemaan adalah apakah  $X$  (sembarang data dengan tipe data sama dengan tipe data yang ada dalam barisan  $A$  dan biasanya di-*input* atau sudah diketahui terlebih dahulu) ada di antara  $A[1] \dots A[n]$  atau apakah  $X$  ada di dalam

atau tidak dengan hasil *Boolean* benar / salah atau sukses / gagal.

Dalam ilmu komputer terdapat bermacam – macam algoritma untuk metoda pencarian (*searching*). Secara garis besar, metoda pencarian data dapat dibagi menjadi 2 bagian yaitu,

1. Metoda pencarian data tanpa penempatan data seperti,
  - a. Metoda pencarian Linier (*Linear / Sequential Search*).
  - b. Metoda pencarian Biner (*Binary Search*).
  - c. Metoda pencarian Interpolasi (*Interpolation Search*).
2. Metoda pencarian data dengan penempatan data seperti,
  - a. Metoda pencarian Langsung (*Direct Search*).
  - b. Metoda pencarian Relatif (Hash Search).

Masing – masing algoritma pencarian memiliki prasyarat dan cara serta waktu pelaksanaan yang berbeda – beda. Pemilihan atas metoda pencarian dilakukan berdasarkan keadaan dan keinginan pengguna metoda yang biasanya tergantung pada jumlah data, jenis data dan struktur data yang digunakan.

Sesuai dengan topik penelitian yang diambil, maka pembahasan metoda pencarian hanya terbatas pada metoda pencarian Linier (*Linear / Sequential Search*), metoda pencarian

Biner (*Binary Search*) dan metoda pencarian Interpolasi (*Interpolation Search*).

### 2.2.1 Metoda Pencarian Linier (**Linear / Sequential Search**)

Pencarian Linier dapat dilakukan pada barisan bilangan yang terurut secara menaik (*ascending*) atau menurun (*descending*) ataupun tidak terurut. Pencarian Linier dilakukan dengan cara membandingkan data yang dicari (X) dengan data dalam barisan  $A[1] \dots A[n]$  dengan dimulai dari data elemen pertama pada barisan A. Jika perbandingan bernilai sama, maka pencarian dihentikan dan dinyatakan sukses. Sedangkan apabila perbandingan tidak bernilai sama maka,

1. Jika data tidak terurut (data acak), maka pencarian akan dilanjutkan ke data selanjutnya.
2. Jika data terurut secara menaik (*ascending*), maka pencarian hanya akan dilanjutkan ke data selanjutnya yang berada di sebelah kanan data yang sedang dibandingkan apabila data yang dicari (X) lebih besar daripada data yang sedang dibandingkan sekarang.
3. Jika data terurut secara menurun (*descending*), maka pencarian hanya akan dilanjutkan ke data selanjutnya yang berada di sebelah kanan data yang sedang dibandingkan apabila data yang dicari (X) lebih kecil daripada data yang sedang dibandingkan sekarang.

Jika syarat – syarat di atas dipenuhi, maka pencarian data akan dilakukan sampai data yang dicari (X) ditemukan sehingga pencarian

dinyatakan sukses atau sampai elemen terakhir dari barisan A dan tidak ada elemen A yang sama dengan data yang dicari (X) sehingga pencarian dinyatakan gagal.

### 2.2.2 Metoda Pencarian Biner (**Binary Search**)

Pencarian Biner hanya dapat dilakukan pada barisan bilangan yang telah diurutkan baik secara menaik (*ascending*) maupun menurun (*descending*). Pencarian Biner melakukan pencarian data X dalam barisan  $A[1] \dots A[n]$  dengan dimulai dari data tengah pada barisan A. Jika nilai data X sama dengan nilai data tengah barisan A, maka pencarian dihentikan dan dinyatakan sukses. Sedangkan jika tidak sama maka,

1. Untuk data yang diurutkan secara menaik (*ascending*), pencarian akan dilanjutkan ke  $\frac{1}{2}$  bagian kiri apabila nilai data X lebih kecil daripada nilai data tengah pada barisan A. Sedangkan apabila nilai data X lebih besar daripada nilai data tengah pada barisan A, maka pencarian akan dilanjutkan ke  $\frac{1}{2}$  bagian kanan.
2. Untuk data yang diurutkan secara menurun (*descending*), pencarian akan dilanjutkan ke  $\frac{1}{2}$  bagian kiri apabila nilai data X lebih besar daripada nilai data tengah pada barisan A. Sedangkan apabila nilai data X lebih kecil daripada nilai data tengah pada barisan A, maka pencarian akan dilanjutkan  $\frac{1}{2}$  bagian kiri.

Pencarian akan dihentikan dan dinyatakan gagal apabila  $\frac{1}{2}$  bagian kiri atau  $\frac{1}{2}$  bagian kanan berupa sebuah data tunggal dan data tersebut tidak sama dengan data X yang sedang dicari.

### 2.2.3 Metoda Pencarian Relatif (Hash Search)

Metoda pencarian Relatif (Hash Search) ini hampir mirip dengan metoda pencarian langsung (*Direct Search*), yaitu dengan menggunakan rumus tertentu baik pada saat penempatan maupun pencarian data. Pencarian Relatif (*Hash Search*) memiliki efisiensi penggunaan tempat yang lebih baik daripada pencarian langsung (*Direct Search*). Fungsi  $(I - 1)$  yang digunakan oleh metoda pencarian langsung (*Direct Search*) memiliki efisiensi penggunaan tempat yang buruk, sehingga metoda pencarian Relatif (*Hash Search*) memperbaikinya dengan menggunakan fungsi operasi modulo ( $\text{mod}$ ). Fungsi operasi modulo ( $\text{mod}$ ) ini sering disebut sebagai Fungsi *Hash* dan tempat penampungan data disebut Tabel *Hash*. Fungsi *Hash* bukan merupakan fungsi satu – satu seperti fungsi bagi dari metoda pencarian Langsung (*Direct Search*) sehingga ada kemungkinan beberapa data memiliki hasil fungsi yang sama. Hal ini mengakibatkan terjadinya tabrakan (*collision*) pada saat penempatan data ke dalam tabel sehingga diperlukan strategi untuk mengatasi tabrakan (*collision*) ini. Strategi untuk mengatasi tabrakan (*collision*) ini ada bermacam – macam dan masing – masing memiliki kelebihan dan kekurangannya masing – masing. Strategi ini akan dijelaskan pada bagian berikutnya.

Fungsi *Hash* ini menyebabkan data yang tersimpan dalam Tabel *Hash* memiliki 2 jenis alamat (*address*) yaitu,

1. *Home Address*, adalah lokasi (*address*) yang diperoleh dengan menggunakan Fungsi *Hash*.
2. *Real (Physical) Address*, adalah lokasi (*address*) dimana data tersimpan dalam tabel.

Metoda pencarian Relatif (*Hash Search*) terdiri dari 2 macam yaitu,

1. Hash Tertutup (*Closed Hash*)
2. Hash Terbuka (*Open Hash*)

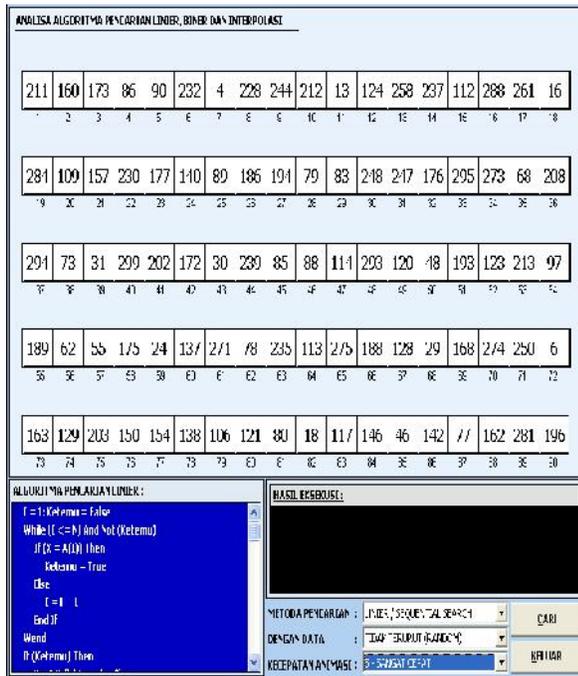
## III. HASIL DAN PEMBAHASAN

### 3.1 Algoritma Simulasi dan Analisa Pencarian

Pada tahap simulasi dan analisa, terdapat prosedur ‘SimulasiCari’, yang berfungsi untuk mensimulasikan tahapan – tahapan pencarian dan menampilkan hasil analisis dan prosedur kerja program dalam bentuk *report*. Prosedur ‘SimulasiCari’ terdiri atas 3 prosedur utama, yaitu ‘*LinearSearch*’, ‘*BinarySearch*’ dan ‘*InterpolationSearch*’. Prosedur ‘*LinearSearch*’ berfungsi untuk mensimulasikan tahapan pencarian dengan algoritma pencarian linier dan menampilkan hasil analisis dan prosedur kerja program dalam bentuk *report*. Prosedur ‘*BinarySearch*’ berfungsi untuk mensimulasikan tahapan pencarian dengan algoritma pencarian biner dan menampilkan hasil analisis dan prosedur

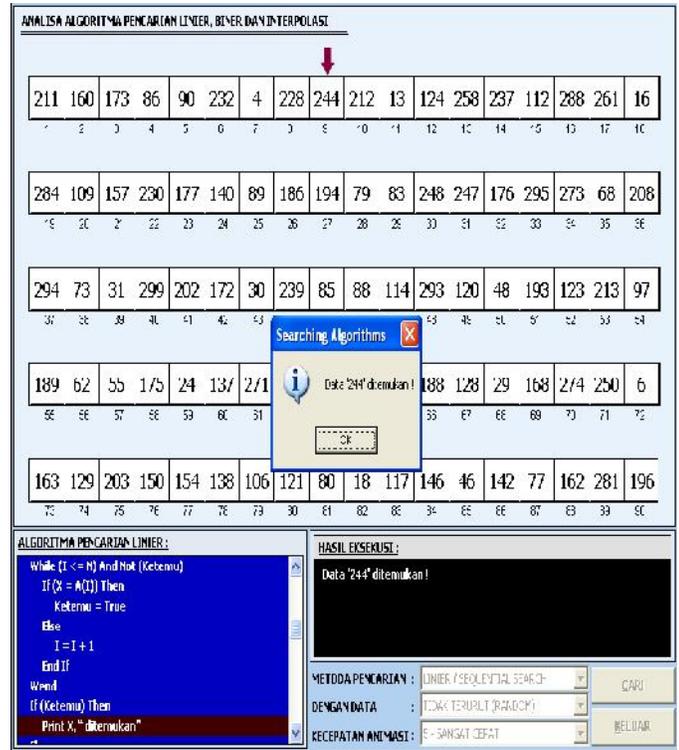
kerja program dalam bentuk *report*. Prosedur 'InterpolationSearch' berfungsi untuk mensimulasikan tahapan pencarian dengan algoritma pencarian interpolasi dan menampilkan hasil analisis dan prosedur kerja program dalam bentuk *report*.

- Proses penempatan barisan data pada tabel, didapat



Gambar 4.1 Tabel Penempatan Data

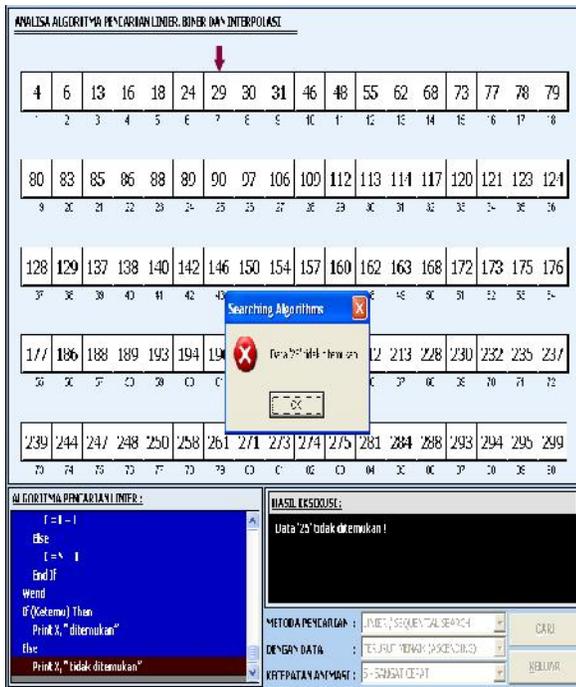
- Pencarian Data '244' dengan metoda pencarian linier (*sequential search*) dan data tidak terurut (*random*), didapat



Gambar 4.2 Hasil Pencarian Data '244'

dengan metoda pencarian linier (*sequential search*) dan data tidak terurut (*random*)

- Pencarian Data '25' dengan metoda pencarian linier (*sequential search*) dan data terurut menaik (*ascending*), didapat



Gambar 4.3 Hasil Pencarian Data '25' dengan metoda pencarian linier (sequential search) dan data terurut menaik (ascending)

- Pencarian Data '288' dengan metoda pencarian linier (sequential search) dan data terurut menurun (descending), didapat

#### IV. KESIMPULAN DAN SARAN

##### 1.1 Kesimpulan

Setelah menyelesaikan perangkat lunak simulasi pergerakan, penulis menarik kesimpulan sebagai berikut:

1. Perangkat lunak menggunakan metode pencarian *breadth first search* (BFS). Karena itu, pencarian akan menemukan semua solusi terpendek.
2. Perangkat lunak merupakan implementasi (penerapan) nyata pohon pelacakan dalam memecahkan suatu permasalahan berdasarkan konsep *Artificial Intelligence* (AI).

3. Perangkat lunak mensimulasikan semua gerakan dari semua solusi yang ditemukan, sehingga memberikan gambaran yang cukup jelas atas solusi-solusi yang dihasilkan.

##### 1.2 Saran

Penulis ingin memberikan beberapa saran yang mungkin dapat membantu dalam pengembangan perangkat lunak ini yaitu:

1. Perangkat lunak dapat dikembangkan dengan menambahkan beberapa metode pencarian lain yang terdapat di dalam ruang lingkup *Artificial Intelligence* (AI).
2. Perangkat lunak dapat dikembangkan dengan menambahkan animasi sewaktu biji kuda digerakkan. Untuk animasi yang lebih baik, perangkat lunak dapat dibangun dengan menggunakan aplikasi *Macromedia Flash*.
3. Perangkat lunak dapat dikembangkan dengan menambahkan biji catur yang lain.

##### DAFTAR PUSTAKA

Arhani.M, Konsep Dasar Sistem Pakar, Penerbit Andi, Yogyakarta, 2005.

Desiani.A dan Arhami.M, Konsep Kecerdasan Buatan, Penerbit Graha Ilmu, 2002.

Kusumadewi.S, Artificial Intelligence (Teknik dan Aplikasinya), Edisi 2, Penerbit Graha Ilmu, 2002.

Munir.R, *Matematika Diskrit*, Informatika Bandung, 2005.

Munir.R, Lidia.L, Algoritma dan Pemrograman,  
Edisi Kedua, 2002.

Pramono.D, Mudah menguasai *Visual Basic 6*, PT.  
Elex Media Komputindo, 2002.

Ramadhan.A , MS. *Visual Basic 6*, PT. Elex  
Media Komputindo, Jakarta, 2004.

Sandi.S, *Artificial Intelegencia*, Andi Offset  
Yogyakarta, 1993.

Supardi.Y, Ir, *Microsoft Visual Basic 6.0 Untuk  
Segala Tingkat*, PT. Elex Media  
Komputindo, Jakarta, 2006.