
APLIKASI ALAT BANTU AJAR ALGORITMA PENCARIAN LINIER, BINER, DAN INTERPOLASI

Oleh :

Hambali Furnawan, ST dan Sulaiman

STMIK Nurdin Hamzah Jambi

Email: hokib411f@yahoo.com,

ABSTRAKSI

Pencarian 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. Agar pencarian dilakukan dengan cepat maka perlu algoritma pencarian, algoritma pencarian yang digunakan meliputi: pencarian linier, pencarian biner dan pencarian interpolasi. Pencarian linier adalah pencarian data secara beruntun mulai dari data pertama sampai dengan data yang dicari ditemukan atau semua data sudah diperiksa. Pencarian biner adalah pencarian data yang dimulai dari pertengahan sekumpulan data yang telah diurutkan berdasarkan data yang dicari. Pencarian interpolasi adalah cara kerja algoritma ini dengan cara menebak posisi data yang dicari dengan menggunakan rumus. Hasil penelitian sudah cukup baik, tetapi penelitian ini belum bias memasukan data yang bertipe karakter. Bahasa yang digunakan dalam pembangunan aplikasi ini yaitu bahasa Microsoft Visual Studio 6.0.

Kata kunci: Algoritam, Biner, Interpolasi, Linier dan Microsoft Visual Studio 6.0.

PENDAHULUAN

Algoritma merupakan salah satu cabang ilmu komputer yang membahas langkah-langkah penyelesaian suatu permasalahan, dengan algoritma yang baik maka komputer bisa menyelesaikan perhitungan dengan cepat dan benar. Baik buruknya sebuah algoritma dapat dibuktikan dari kompleksitas waktu yang digunakan. Terdapat bermacam – macam algoritma untuk pencarian (*searching*). 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.

Beberapa pencarian yang akan dianalisa dalam penelitian ini adalah 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 pencarian dilakukan berdasarkan keadaan dan keinginan pengguna yang biasanya tergantung pada jumlah data, jenis data dan struktur data yang digunakan. Data yang akan diproses berupa data numerik yang kemudian dari sekumpulan data tersebut akan dicari nilai numerik yang diinginkan. Pencarian dilakukan pada sekumpulan data yang diinput secara random untuk kemudian pencarian dapat dilakukan pada data random tersebut atau data diurutkan secara *ascending* (terurut menaik) atau secara *descending* (terurut menurun). Untuk memudahkan pencarian, digunakan algoritma pengurutan data *bubble sort*.

1. LANDASAN TEORI

2.1 Algoritma

Algoritma adalah suatu kumpulan terhingga (*finite set*) dari instruksi yang terdefinisi dengan baik (*well-defined instructions*) untuk menyelesaikan beberapa pekerjaan dimana diberikan *state* awal (*initial state*) dan akan dihentikan pada saat ditemukan *state* akhir (*end-state*). Algoritma dapat diimplementasikan dalam pembuatan program komputer. Kesalahan

dalam merancang algoritma untuk menyelesaikan suatu problema dapat menyebabkan program gagal dalam implementasinya.

Konsep dari suatu algoritma sering diilustrasikan dengan mengambil contoh sebuah resep, walaupun banyak algoritma yang jauh lebih kompleks. Algoritma sering memiliki beberapa langkah perulangan (iterasi) atau memerlukan pengambilan keputusan seperti logika (*logic*) atau perbandingan (*comparison*) sampai pekerjaan diselesaikan. Menerapkan suatu algoritma secara benar belum tentu dapat menyelesaikan problema. Hal ini dikarenakan adanya kemungkinan algoritma tersebut rusak atau cacat, atau penerapannya tidak cocok (tidak tepat) untuk menyelesaikan problema. Sebagai contoh, sebuah algoritma hipotesis untuk membuat sebuah salad kentang akan gagal jika tidak terdapat kentang. Suatu pekerjaan dapat diselesaikan dengan menggunakan algoritma yang berbeda dengan kumpulan instruksi (*set of instructions*) yang berbeda dengan perbedaan waktu akses, efisiensi tempat, usaha dan sebagainya.

2.2 Algoritma Sorting

Untuk pengolahan dan memudahkan pencarian data dari dalam suatu arsip atau struktur data maka diperlukan data yang sudah berurutan, bertujuan *sorting* adalah untuk mempercepat pencarian dan kerapian susunan data. Pada aplikasi ini penulis menggunakan metode *bubble sort*, prinsip kerja algoritma *bubble sort* seperti gelembung udara, dimana gelembung tersebut akan naik terus menerus sampai diperoleh posisi tertinggi dipermukaan air, dalam algoritma ini bilangan akan dibandingkan dari posisi paling bawah dengan bilangan di atasnya. Jika bilangan pada *array* kanan lebih kecil dari bilangan *array* sebelah kiri maka isi *array* akan ditukar tempatnya. Iterasi ini akan dijalankan sampai semua bilangan yang diurutkan pada posisi yang benar.

Algoritma Bubble sort

For $i=1$ to $N-1$ do

For $k=N$ down to $i+1$ do

If $L[K] < L[K-1]$ then

$Tem=L[K]$

$L[K]=L[K-1]$

$L[K-1]=tem$

2.3 Algoritma 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.

2.3.1 Pencarian Linier

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.

Metoda pencarian Linier untuk data tidak terurut (data acak) dalam bahasa pemrograman *Basic* adalah sebagai berikut,

```
I = 1: Ketemu = False
While (I <= N) And Not (Ketemu)
    If (X = A(I)) Then
        Ketemu = True
    Else
        I = I + 1
    End If
Wend
If (Ketemu) Then
    Print X, " ditemukan "
Else
    Print X, " tidak ditemukan "
```

Sebagai contoh kasus, misalkan diketahui suatu barisan bilangan A sebagai berikut,

12	7	1	3	9	19	25	17
1	2	3	4	5	6	7	8

Misalkan data yang dicari (X) = 3 maka proses pencarian data dengan menggunakan metoda pencarian Linier untuk data tidak terurut (data acak) adalah sebagai berikut :

1. Pencarian dimulai dari data elemen pertama pada barisan bilangan A.

↓

12	7	1	3	9	19	25	17
1	2	3	4	5	6	7	8

$$N = 8: X = 3$$

$$I = 1: \text{Ketemu} = \text{False}$$

$$\text{While } (1 \leq 8) \text{ And Not (False)} \rightarrow \text{True}$$

$$\text{If } (3 = 12) \text{ Then } \rightarrow \text{False}$$

$$I = 1 + 1 = 2$$

2. Data tidak ditemukan, pencarian dilanjutkan ke data elemen kedua pada barisan bilangan A.

↓

12	7	1	3	9	19	25	17
1	2	3	4	5	6	7	8

$$\text{While } (2 \leq 8) \text{ And Not (False)} \rightarrow \text{True}$$

$$\text{If } (3 = 7) \text{ Then } \rightarrow \text{False}$$

$$I = 2 + 1 = 3$$

3. Data tidak ditemukan, pencarian dilanjutkan ke data elemen ketiga pada barisan bilangan A.

		↓					
12	7	1	3	9	19	25	17
1	2	3	4	5	6	7	8

While ($3 \leq 8$) And Not (False) \rightarrow True

If ($3 = 1$) Then \rightarrow False

$I = 3 + 1 = 4$

4. Data tidak ditemukan, pencarian dilanjutkan ke data elemen keempat pada barisan bilangan A.

			↓				
12	7	1	3	9	19	25	17
1	2	3	4	5	6	7	8

While ($4 \leq 8$) And Not (False) \rightarrow True

If ($3 = 3$) Then \rightarrow True

Ketemu = True

While ($4 \leq 8$) And Not (True) \rightarrow False

If (True) Then

Print 3, “ditemukan”

5. Data ditemukan pada posisi keempat dari barisan bilangan A dan pencarian sukses.

2.3.2 Pencarian Biner

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.

Metoda pencarian Biner untuk data terurut secara menaik (*ascending*) dalam bahasa pemrograman *Basic* adalah sebagai berikut,

Kiri = 1: Kanan = n: Ketemu = False

While (Kiri <= Kanan) And Not (Ketemu)

Tengah = (Kiri + Kanan) Div 2

If (X = A(Tengah)) Then

Ketemu = True

Else If (X < A(Tengah)) Then

Kanan = Tengah - 1

Else

Kiri = Tengah + 1

End If

Wend

If (Ketemu) Then

Print X, “ ditemukan”

Else

Print X, “ tidak ditemukan”

Misalkan data yang dicari (X) = 3 maka proses pencarian data dengan menggunakan metoda pencarian Biner untuk data terurut secara menaik (*ascending*) adalah sebagai berikut,

Untuk metoda pencarian Biner diambil contoh barisan bilangan A yang sama seperti contoh pada pencarian Linier di atas sebagai berikut,

12	7	1	3	9	19	25	17
1	2	3	4	5	6	7	8

Barisan bilangan A diurutkan secara menaik (*ascending*) seperti berikut in

1	3	7	9	12	17	19	25
1	2	3	4	5	6	7	8

1. Pencarian dimulai dari data tengah pada barisan bilangan A.



1	3	7	9	12	17	19	25
1	2	3	4	5	6	7	8

Kiri = 1: Kanan = 8: Ketemu = False

While ($1 \leq 8$) And Not (False) \rightarrow True

Tengah = $(1 + 8) \text{ Div } 2 = 4$

If ($3 < 9$) Then

Kanan = $4 - 1 = 3$

2. Data tidak ditemukan dan nilai data X yang dicari lebih kecil daripada nilai data tengah pada barisan bilangan A, sehingga pencarian dilanjutkan ke ½ bagian kiri.

	↓						
1	3	7	9	12	17	19	25
1	2	3	4	5	6	7	8

While (1 <= 3) And Not (False) → True

Tengah = (1 + 3) Div 2 = 2

If (3 = 3) Then

Ketemu = True

While (1 <= 3) And Not (True) → False

If (True) Then

Print 3, “ ditemukan”

3. Data ditemukan pada posisi kedua dari barisan bilangan A dan pencarian sukses

2.3.3 Metoda Pencarian Interpolasi (Interpolation Search)

Pencarian Interpolasi hanya dapat dilakukan pada barisan bilangan yang telah diurutkan baik secara menaik (*ascending*) maupun menurun (*descending*). Posisi yang diduga sebagai posisi data yang dicari (X) tersebut dapat dihitung dengan menggunakan rumus berikut.

$$\text{Pos} = \text{ceiling} [\text{BB} + ((X - A[\text{BB}]) / (A[\text{BA}] - A[\text{BB}])) * (\text{BA} - \text{BB})]$$

Keterangan :

Pos = posisi yang diduga

BB = Batas Bawah; mula-mula

sama dengan 1

BA = Batas Atas; mula-mula sama dengan jumlah data (n) X = Data yang dicari

Ceiling = operasi matematika untuk pembulatan angka ke atas

Apabila data pada posisi tersebut sama dengan nilai data yang dicari (X) maka pencarian dihentikan dan dinyatakan sukses. Sedangkan apabila nilai data pada posisi yang diduga tidak sama dengan nilai data yang dicari (X), maka :

1. Untuk data yang diurutkan secara menaik (*ascending*), apabila nilai data yang dicari (X) < dibandingkan dengan nilai data pada posisi yang diduga, maka pencarian dilanjutkan dengan mengubah batas atas kawasan pencarian. Sedangkan apabila nilai data yang dicari (X) > dibandingkan dengan nilai data pada posisi yang diduga, maka pencarian dilanjutkan dengan mengubah batas bawah kawasan pencarian.
2. Untuk data yang diurutkan secara menurun (*descending*), apabila nilai data yang dicari (X) > dibandingkan dengan nilai data pada posisi yang diduga, maka pencarian dilanjutkan dengan mengubah batas atas kawasan pencarian. Sedangkan apabila nilai data yang dicari (X) < dibandingkan dengan nilai data pada posisi yang diduga, maka pencarian dilanjutkan dengan mengubah batas bawah kawasan pencarian.

Proses pencarian akan dihentikan dan dinyatakan gagal apabila terjadi kondisi – kondisi berikut ini,

- a) Nilai Pos < nilai batas bawah b) Nilai Pos > nilai batas atas c) Nilai posisi dugaan berturut – turut sama.

Misalkan contoh kasus, data yang dicari (X) = 3, data terurut secara menurun (*descending*)

25	19	17	12	9	7	3	1
1	2	3	4	5	6	7	8

1. Hitung posisi yang diduga sebagai posisi data yang dicari (X).

$$BB = 1: BA = 8: Ketemu = False: CekPos = 0$$

While ($1 \leq 8$) And Not (False) \rightarrow True

$$\text{Temp} = 1 + ((3 - 25) / (1 - 25)) * (8 - 1) = 7,416667$$

$$\text{Pos} = \text{INT}(7,416667) = 7$$

If ($7,416667 - 7 > 0$) Then \rightarrow True

$$\text{Pos} = 7 + 1 = 8$$

If ($3 > 1$) Then

$$\text{BA} = 8 - 1 = 7$$

25	19	17	12	9	7	3	1
1	2	3	4	5	6	7	8

↓

2. Nilai data elemen kedelapan pada barisan bilangan A tidak sama dengan nilai data yang dicari (X), sehingga pencarian dilanjutkan dengan menghitung nilai posisi diduga berikutnya.

While ($1 \leq 7$) And Not (False) \rightarrow True

$$\text{Temp} = 1 + ((3 - 25) / (3 - 25)) * (7 - 1) = 7$$

$$\text{Pos} = \text{INT}(7) = 7$$

If ($7 - 7 > 0$) Then \rightarrow False

If ($3 = 3$) Then

Ketemu = True

While ($1 \leq 7$) And Not (True) \rightarrow False

If (True) Then

Print 3, “ ditemukan”

25	19	17	12	9	7	3	1
1	2	3	4	5	6	7	8

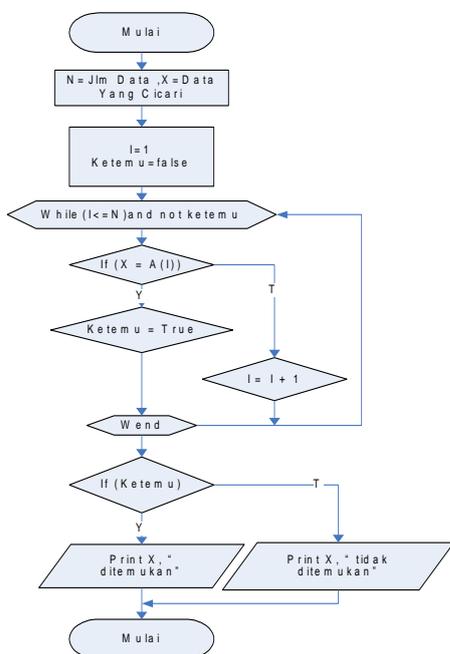
↓

3. Nilai data elemen ketujuh pada barisan bilangan A sama dengan nilai data yang dicari (X), sehingga pencarian dihentikan dan dinyatakan sukses.

2. DIAGRAM ALIR SISTEM

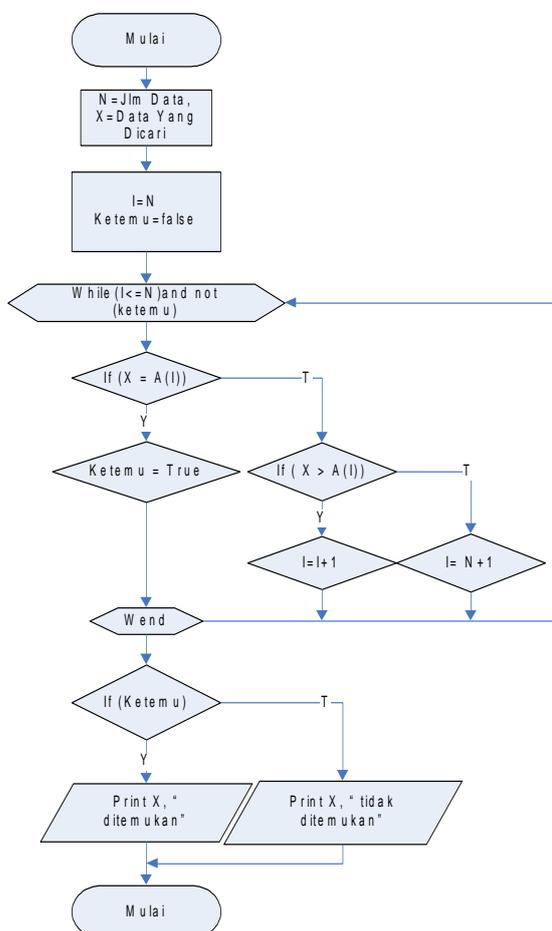
3.1 Perancangan Diagram Alir Metode

Linier Data Acak (*Random*)



Gambar 1 Diagram Alir Metode Linier Data Acak

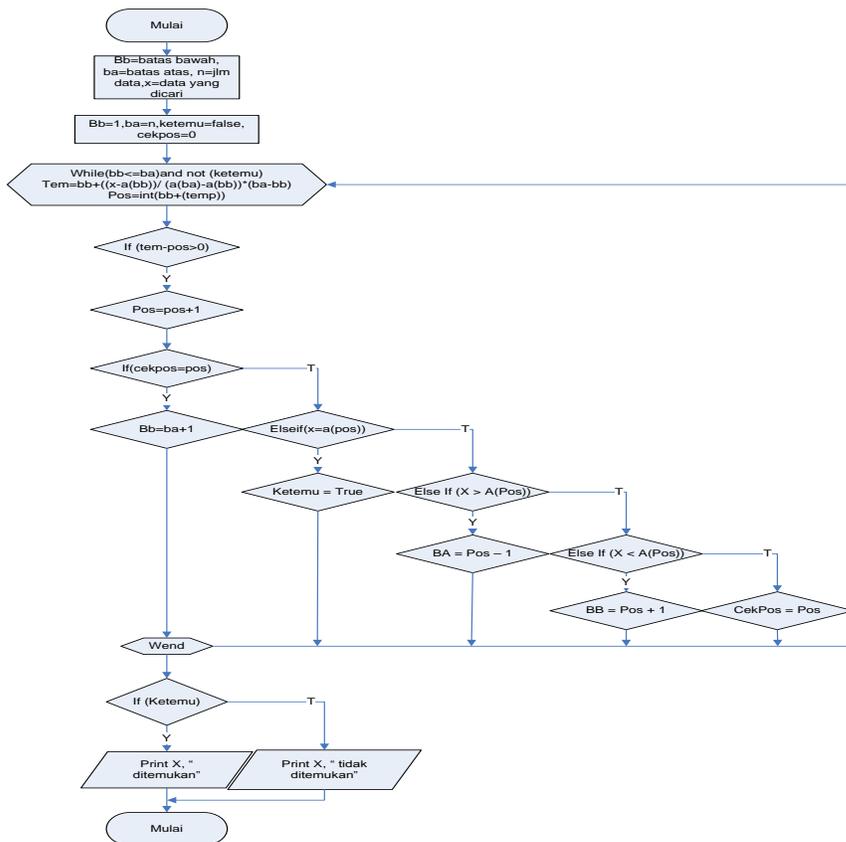
3.2 Perancangan Diagram Alir Metode Biner Data Ascending



Gambar 2 Diagram Alir Metode Biner Data Ascending

3.3 Perancangan Diagram Alir Metode Interpolasi Data

Descending

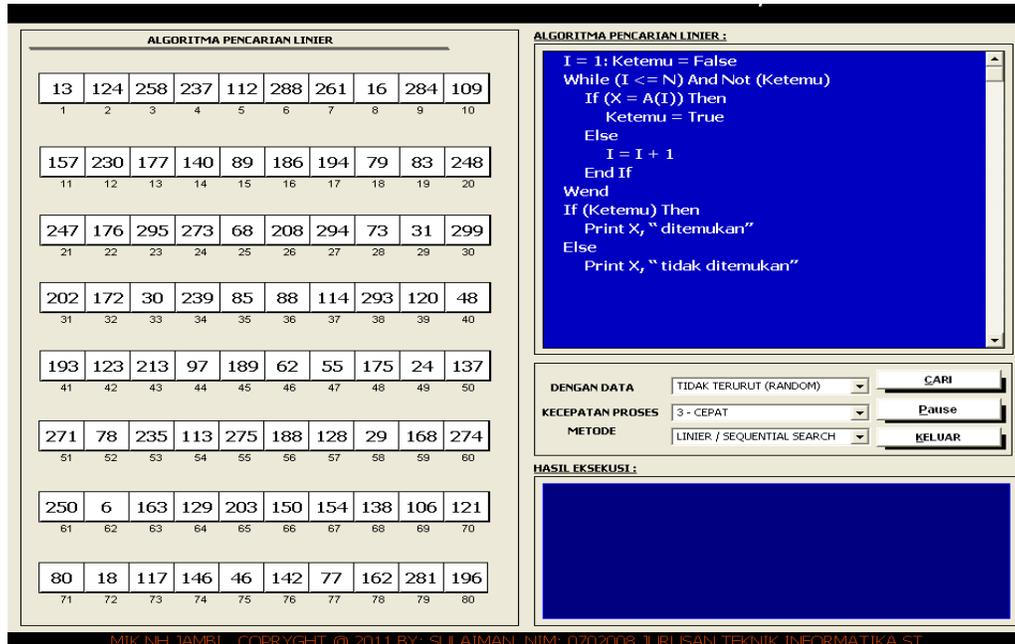


Gambar 3 Diagram Alir Metode Interpolasi Data Descending

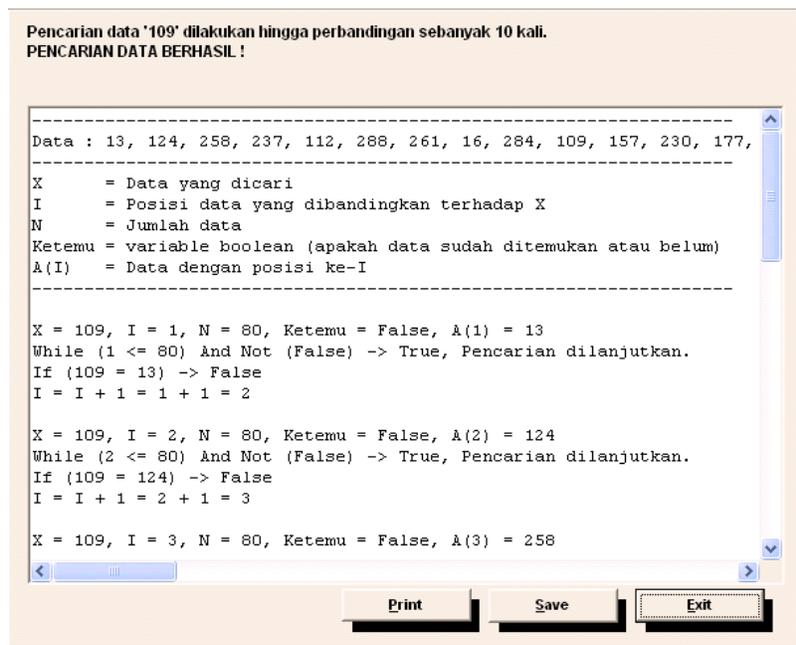
4. IMPLEMENTASI DAN PENGUJIAN



Gambar 3 Antarmuka Menu *Input Data*



Gambar 4. Antarmuka Menu Pencarian



Gambar 5. Antarmuka Langkah-langkah Hasil Pencarian

5. KESIMPULAN DAN SARAN

Aplikasi alat bantu ajar algoritma linier, biner dan interpolasi ini dapat digunakan sebagai salah satu alternatif untuk mempermudah pembelajaran mahasiswa dalam mengetahui langkah-langkah pencarian berdasarkan algoritma linier, biner, dan interpolasi dari 3 algoritma pencarian (*searching*) tersebut secara langkah per langkah. Untuk pengembangan selanjutnya, dapat disertakan analisa tentang kompleksitas waktu pencarian serta besarnya memori yang dibutuhkan oleh masing-masing algoritma. Selain itu, metode pengurutannya dapat dikembangkan dengan metode pengurutan yang lain seperti *heap sort, selection sort, insertion sort, shell sort, marge sort*.

DAFTAR PUSTAKA

Ada Bryan, Pengenalan Algoritma dan qbasic, Bandung, Informatika, 1953.

Amunsend dan Curtis, S: Dasar-dasar *Pemrograman Database* dan *Visual Basic*, Yogyakarta: Andi, 1997.

Budi Sutedjo,, S.Kom dan Michael An: Algoritma dan Teknik Pempograman, Yogyakarta: Andi, 1997.

Eko Budi Purwanto, rancangan dan Analisa Algoritma, Yogyakarta: Graha Ilmu,2008.

Ema Utami,S.Si., M.Kom dan Suwanto Raharjo,S.Si., M.Kom: Logika Algoritma dan Implementasinya dalam Bahasa Python di GNU/Linux, yogyakarta: Andi, 2004.

Fajar Junaedi, Pengantar Algoritma dan pemrograman, Jakarta: Salemba Infotek, 2007

Galih Pranowo, Desain Analisis dan Algoritma, yogyakarta, Andi, 2004.

Hedlin Bahtiar, Algorithm and Programing,Bandung, Informatika, 2002.

Reny Wahyuning Astuti, Modul Praktikum *Pemrograman Visual Basic*, Jambi: Pusal Laboratorium Komputer STMIK Nurdin Hamzah, 2008