

ANALISA ALGORITMA BINARY SEARCH UNTUK Mencari DATA MAHASISWA PROGRAM STUDI MANAJEMEN UNIVERSITAS PELITA BANGSA DENGAN BERBASIS ANDROID

Yoga Religia¹, Ahmad Nurhakim²

Program Studi Teknik Informatika Universitas Pelita Bangsa
yoga.religia@pelitabangsa.ac.id

Disetujui, 31 Agustus 2019

ABSTRAK

Pencarian merupakan pekerjaan yang sering kita lakukan dalam kehidupan sehari-hari. Dalam text editor, kita sering melakukan pekerjaan mencari kata, atau mencari kata1 dan menggantikannya dengan kata 2. Tujuan dari penelitian ini untuk mengetahui analisis kecepatan rata rata waktu proses pencarian dan ketepatan algoritma *sequential search* untuk mencari data mahasiswa teknik informatika pada Universitas Pelita Bangsa. Metode yang digunakan adalah Algoritma *Binary Search*. Hasil dari penelitian ini dapat diambil kesimpulan yaitu algoritma *searching* merupakan algoritma yang penting dalam pengelolaan sistem manajemen *database*. Data yang besar (*Big Data*) perlu diolah untuk mempermudah dalam pencarian data. Analisis kecepatan rata - rata waktu pencarian dan ketepatan algoritma *binary search* diketahui dengan perhitungan kompleksitas waktu hasil dari penelitian ini diketahui analisis kecepatan rata- rata waktu 0,025 ms dalam jumlah data mahasiswa untuk membantu proses analisis sebanyak 2000 data berupa nim dan nama mahasiswa. Pengujian dalam penelitian ini menggunakan analisa kompleksitas waktu pencarian, analisa kompleksitas dengan notasi linear $O(n)$, dan *black box testing*. *Black box testing* hasil pengujian sesuai dan *valid*.

Kata kunci: *Binary Search, Sequential Search, Black Box Testing, Asimptotik, Universitas Pelita Bangsa*.

ABSTRACT

Searching is a job that we often do in our daily lives. In text editors, we often do the work of searching for words, or searching for words1 and replacing them with words 2. The purpose of this study is to find out the analysis of the average speed of the search process and the accuracy of sequential search algorithms to look for data of informatics engineering students at Pelita Bangsa University. The method used is the Binary Search Algorithm. The results of this study can be concluded that the searching algorithm is an important algorithm in managing a database management system. Big data (Big Data) needs to be processed to make it easier to find data. Analysis of the average speed of the search time and the accuracy of the binary search algorithm is known by calculating the time complexity of the results of this research is known to an analysis of the average speed of time in 0.025 ms the number of student data to assist the analysis process is 2000 data in the form of student numbers and names. Tests in this study use the analysis of search time complexity, complexity analysis with linear $O(n)$ notation, and black box testing. Black box testing testing results are appropriate and valid.

Kata kunci: *Binary Search, Sequential Search, Black Box Testing, Asimptotik, Universitas Pelita Bangsa*.

I Pendahuluan

Pencarai data merupakan suatu pekerjaan yang terbilang mudah tetapi jika dilakukan secara manual ataupun tidak menggunakan cara sistem khusus pencarian data maka pekerjaan dalam pencarian data akan terbilang sulit dan bisa menghabiskan waktu yang cukup lama karena data yang dicari terbilang banyak. Misalkan dalam penelitian ini akan menggunakan data mahasiswa Universitas Pelita Bangsa program studi manajemen dan dalam penelitian saat ini akan di terapkannya dengan cara metode sistem pencarian data menggunakan algoritma *binary search* yang dimana metode ini adalah salah satu metode sistem proses pencarian data, seberapa cepat algoritma *binary search* dalam proses mencari data mahasiswa program studi manajemen Universitas Pelita Bangsa.

Pencarian merupakan pekerjaan yang sering kita lakukan dalam kehidupan sehari-hari. Dalam text editor, kita sering melakukan pekerjaan mencari kata, atau mencari kata 1 dan menggantikannya dengan kata 2. Metode pencarian (*searching*) terbagi lima, yaitu metode pencarian linier (*Linear / Sequential Search*), metode pencarian biner (*Binary Search*), metode pencarian interpolasi (*Interpolation Search*), metode pencarian langsung (*Direct Search*), metode pencarian relatif (*Hash Search*). Masing-masing metode memiliki kelebihan dan kekurangan. Salah satu metode pencarian (*searching*) yang memiliki efisiensi adalah metode pencarian biner (*Binary Search*). Pencarian biner (*Binary Search*) dapat dilakukan pada barisan bilangan yang telah diurutkan, baik secara menaik (*ascending*) maupun secara menurun (*descending*). Tujuan dari penelitian ini untuk mengetahui analisis kecepatan rata-rata waktu proses pencarian dan ketepatan algoritma *binary search* untuk mencari data mahasiswa program studi manajemen pada Universitas Pelita Bangsa.

2. Tinjauan Studi

Setelah melakukan analisa terhadap beberapa penelitian, ada beberapa penelitian yang memiliki keterkaitan dengan penelitian yang kami lakukan.

2.1 Implementasi Rule Based Dan Binary Untuk Pencarian Huruf Karakter

Latin Pada Aplikasi Transliterasi Latin - Arab

Tujuan dalam penelitian ini yaitu menerapkan metode *Rule Based* dan *Binary Search* untuk mengukur tingkat akurasi dalam aplikasi *Transliterasi* Latin-Arab. Hasil yang dicapai dengan tingkat akurasi yang diperoleh mencapai 100% dengan input kata maupun secara percobaan sebanyak 260 kali dan rata-rata waktu 0.016 detik dan tingkat akurasi transliterasi dari arab ke latin rata-rata yang diperoleh mencapai 99.6% percobaan sebanyak 50 kali dan rata-rata waktu 0.026 detik.

2.2 Pengertian Algoritma

Algoritma adalah deretan instruksi yang jelas untuk memecahkan masalah, yaitu untuk memperoleh keluaran yang diinginkan dari suatu masukan. Ada 3 definisi tentang algoritma yang dijelaskan diantaranya :

- 1) Teknik penyusunan langkah-langkah penyelesaian masalah dalam bentuk kalimat dengan jumlah kata terbatas tetapi tersusun secara logis dan sistematis.
- 2) Suatu prosedur yang jelas untuk menyelesaikan suatu persoalan dengan menggunakan langkah-langkah tertentu dan terbatas jumlahnya.
- 3) Susunan langkah yang pasti, yang bila diikuti maka akan mentransformasi data input menjadi output yang berupa informasi.

Algoritma merupakan suatu prosedur untuk menyelesaikan suatu masalah yang tersusun secara logis dan sistematis serta akan memperoleh data masukan menjadi keluaran yang diinginkan berupa informasi

2.3 Pengertian Binary Search

Binary Search adalah algoritma pencarian yang lebih efisien daripada algoritma *Sequential Search*. Hal ini dikarenakan algoritma ini tidak perlu menjelajahi setiap elemen dari table. Kerugiannya algoritma ini hanya biasa digunakan pada tabel yang elemennya sudah terurut baik menaik maupun menurun.

Binary Search merupakan salah satu algoritma untuk melakukan pencarian pada array yang sudah terurut. Namun jika kita telah mengetahui integer dalam array terorganisasi baik secara menaik maupun menurun maka bisa dengan cepat menggunakan algoritma *Binary Search*. Adapun ide dasar *Binary Search* yaitu memulai pencarian dengan membagi dua ruang pencarian.

3. Desain Penelitian/Metodologi Metode Penelitian

3.1 Jenis Data Yang Digunakan

Data sekunder pada penelitian ini menggunakan teknik pengumpulan data berupa Studi Pustaka. Studi Pustaka yang dilakukan memiliki acuan pada buku-buku ahli serta jurnal maupun literatur lainnya yang memiliki cakupan penelitian sejenis untuk mendukung teori yang digunakan serta menjadikan perbandingan dengan hasil yang didapat dari penelitian.

3.2 Metode yang digunakan

Kompleksitas dari suatu algoritma merupakan ukuran seberapa banyak komputasi yang dibutuhkan algoritma untuk menyelesaikan masalah. Algoritma yang dapat menyelesaikan suatu permasalahan dalam waktu yang singkat memiliki kompleksitas yang rendah, sementara algoritma yang membutuhkan waktu lama untuk menyelesaikan masalah mempunyai kompleksitas yang tinggi.

Setiap algoritma memiliki dua buah ciri khas yang dapat digunakan sebagai parameter perbandingan, yaitu jumlah proses yang dilakukan dan jumlah memori yang digunakan untuk melakukan proses. Jumlah proses ini dikenal sebagai kompleksitas waktu yang disimbolkan dengan $T(n)$, diukur dari jumlah tahapan komputasi yang dibutuhkan untuk menjalankan algoritma sebagai fungsi dari ukuran masukan n , dimana ukuran masukan (n) merupakan jumlah data yang diproses oleh sebuah algoritma. Sedangkan jumlah memori dikenal sebagai kompleksitas ruang yang disimbolkan dengan $S(n)$, diukur dari memori yang digunakan oleh struktur data yang terdapat didalam algoritma sebagai fungsi dari masukan.

3.3 Langkah-langkah Penelitian

Langkah-langkah yang digunakan dalam penelitian ini adalah sebagai berikut:

- 1) Penelitian Kepustakaan Penelitian Kepustakaan dilakukan dengan mengumpulkan dan mempelajari literatur yang berkaitan dengan metode SAW dan TOPSIS untuk pemecahan masalah yang diterapkan dalam model *decision* DSS. Sumber literatur berupa buku teks, paper, journal, karya ilmiah, dan situs-situs penunjang. Kegunaan metode ini diharapkan dapat mempertegas teori serta keperluan analisis dan mendapatkan data yang sesungguhnya
- 2) Observasi, Observasi dilakukan dengan mengumpulkan data dengan melakukan pengamatan secara langsung dan sistematis terhadap objek atau proses yang terjadi.
- 3) Wawancara, Wawancara dilakukan untuk mencari data dan informasi tentang hal-hal yang dibutuhkan dalam penelitian. Wawancara dilakukan dengan manager pada divisi-divisi. Wawancara yang dilakukan lebih menitikberatkan bagaimana prosedur pemilihan karyawan kontrak yang akan menjadi karyawan tetap.

3.4 Analisis Kebutuhan

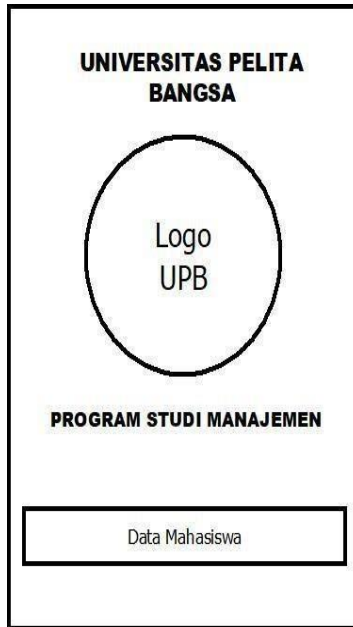
Pada tahap ini dijelaskan analisa masalah, analisa kebutuhan sistem, analisa perangkat lunak dan perangkat keras, analisa kebutuhan pengguna.

- 1) Analisa kebutuhan menggunakan algoritma *binary search* untuk memperoleh hasil yang diinginkan yaitu diketahui kecepatan rata-rata waktu proses pencarian dan kemangkusan dari algoritma pencarian tersebut.
- 2) Analisa kebutuhan sistem menjelaskan gambaran umum dari sistem yang di buat, analisa kebutuhan fungsional yang dibutuhkan untuk mempermudah analisis algoritma *binary search* dalam pencarian data mahasiswa antara lain :
 - a) Sistem dapat menampilkan pencarian data mahasiswa fakultas manajemen Universitas Pelita Bangsa.
 - b) Sistem dapat menampilkan waktu proses untuk mencari data mahasiswa fakultas manajemen Universitas Pelita Bangsa.
- 3) Analisa kebutuhan perangkat lunak yang digunakan dalam pembuatan aplikasi untuk membantu analisa mencari data mahasiswa pada penelitian ini adalah :
 - a) Sistem operasi : windows 10
 - b) Bahasa pemrograman : Java
 - c) Pembuatan Aplikasi : Android Studio Versi 3.5.3

d) Database : SQLite

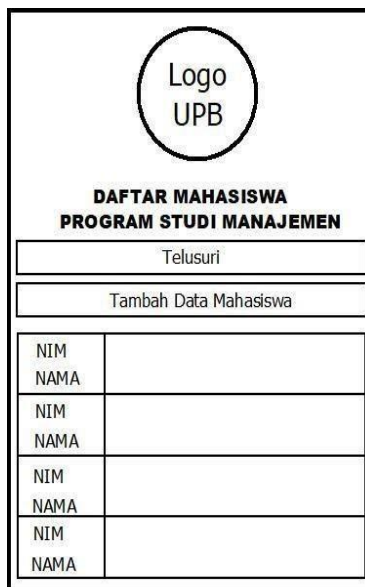
1. Desain Aplikasi

Desain aplikasi tahapan ini dilakukan untuk menetapkan bagaimana sistem akan dioperasikan. Hal ini berkaitan dengan menentukan tampilan program yang akan digunakan serta menampilkan database. Pada tampilan desain aplikasi terdiri dari text field untuk mengetik data yang dicari, button untuk tombol pencarian data, dan table untuk menampilkan data mahasiswa yang di cari. Berikut tampilan desain aplikasi yang digunakan untuk mencari data mahasiswa program studi manajemen pada Universitas Pelita Bangsa:



Gambar 3.1 Tampilan Halaman Utama

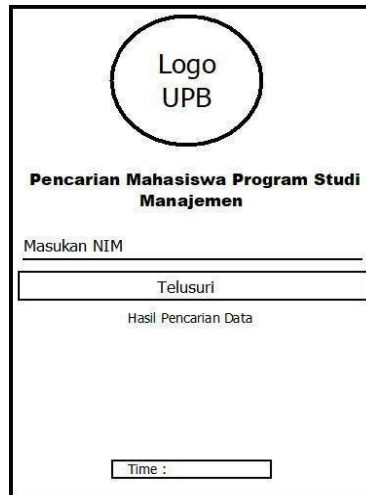
Gambar 3.1 merupakan tampilan awal ini hanya ada tombol data mahasiswa yang dimana tombol ini untuk menampilkan halaman data mahasiswa program studi manajemen Universitas Pelita bangsa.



Gambar 3.2 Tampilan Data Mahasiswa

Gambar 3.2 merupakan tampilan data mahasiswa fakultas manajemen yang didalamnya

terdapat daftar mahasiswa. Di halaman ini juga ada tombol telusur yang fungsinya untuk mencari data mahasiswa dan ada juga tombol tambah data yang fungsinya untuk menambah data mahasiswa.



Gambar 3. 3 Tampilan Penelusuran

Tabel 3 . 1 Contoh Pengujian Black Box

Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
Menulis NIM dan tekan tombol telusur “i”	Menampilkan data mahasiswa yang dicari dan <i>run time</i> pencarian	Sesuai/tidak sesuai	Valid/tidak valid

2. Black Box Testing

Metode pengujian *Black Box* merupakan pengujian suatu perangkat lunak atau sistem dengan menguji secara fungsional berdasarkan pada spesifikasi kebutuhan perangkat lunak. Metode uji dapat diterapkan pada semua tingkat pengujian perangkat lunak, unit, integrasi, fungsional, sistem dan penerimaan. Pengujian pada Black Box berusaha menemukan kesalahan diantaranya:

1. Kesalahan dari inialisasi dan terminasi
2. Kesalahan kinerja
3. Kesalahan dari struktur data atau akses database
4. Kesalahan *interface*
5. Fungsi – fungsi yang tidak benar atau hilang

3.5. Langkah Langkah Algoritma

Mula-mula diambil posisi awal 0 dan posisi akhir = N - 1, kemudian dicari posisi data tengah dengan rumus (posisi awal + posisi akhir) / 2. Kemudian data yang dicari dibandingkan dengan data tengah. Jika lebih kecil, proses dilakukan kembali tetapi posisi akhir dianggap sama dengan posisi tengah - 1. Jika lebih besar, proses dilakukan kembali tetapi posisi awal dianggap sama dengan posisi tengah + 1. Demikian seterusnya sampai data tengah sama dengan yang dicari.

Tabel 3 . 2 Contoh Ilustrasi Binary Search 1

Urutkan terlebih dahulu semua bilangan, boleh terurut menaik atau terurut menurun. Kali ini saya akan menggunakan terurut menaik sehingga diperoleh array sebagai berikut dan kemudian saya tambah dengan keterangan urutan di baris bawah.

Tabel 3 . 3 Contoh Ilustrasi Binary Search 2

Nilai	1	2	4	4	6	7	9	23	25	35
Index	1 (IB)	2	3	4	5	6	7	8	9	10 (IA)

2. Tentukan Indeks atas (Ia) dan indeks bawah (Ib). Ia adalah nilai indeks terbesar dari array yaitu 10. Sedangkan Ib indeks terkecil dari array yaitu 1. dan kita tetapkan bahwa saat ini ketemu=tidak ketemu.
3. Cek apakah saat ini ketemu != ketemu dan Ib < dari Ia? Jika iya maka buka elemen ke-k, dimana k adalah $Ia+Ib/2 = 10+1/2= 5$ (Biasanya pembulatan kebawah).
4. Sekarang periksa elemen ke-5 apakah sama dengan 2 (bilangan yang dicari)? Ternyata bukan, elemen ke-5 adalah 6.
5. Karena bukan maka periksa apakah nilai elemen ke-5 > dari 2?? Jika iya maka nilai Ia akan diubah menjadi k atau sama saja dengan $Ia=k=5$. Sehingga area yang dicari menjadi lebih sempit.

Tabel 3 . 3 Contoh Ilustrasi Binary Search 3

Nilai	1	2	4	4	6
Index	1 (IB)	2	3	4	5 (IA)

1. Cek apakah saat ini ketemu != ketemu dan Ib < dari Ia? Jika iya maka buka elemen ke-k, dimana k adalah $Ia+Ib/2 = 1+5/2= 3..$
2. periksa elemen ke-3 apakah sama dengan 2 (bilangan yang dicari)? Ternyata bukan, elemen ke-3 adalah 4.
3. Karena bukan maka periksa apakah nilai elemen ke-3 > dari 2?? Jika iya maka nilai Ia akan diubah menjadi k atau sama saja dengan $Ia=k=3$.

Tabel 3 .4 Contoh Ilustrasi Binary Search 4

Nilai	1	2	4
Index	1 (IB)	2	3 (IA)

1. Cek apakah saat ini ketemu != ketemu dan Ib < dari Ia? Jika iya maka buka elemen ke-k, dimana k adalah $Ia+Ib/2 = 1+3/2= 2$.
2. periksa elemen ke-2 apakah sama dengan 2 (bilangan yang dicari)? Ternyata benar. Jika benar ketemu =ketemu.

1. Contoh Perhitungan Kompleksitas Waktu Pencarian
 - a. $T_{min}(n)$: kompleksitas waktu untuk kasus terbaik (*best case*) -> kebutuhan waktu minimum.
 $T_{min}(n) = 1$
 - b. $T_{max}(n)$: kompleksitas waktu untuk kasus terburuk (*worst case*) -> kebutuhan waktu maksimum. $T_{max}(n) = 2 \log n$

```

Kamus
Const N : integer = 8 { misalkan jumlah elemen array maksimum = 8 }
Type A = array [ 1 ..... N ] of integer
Cari, BatasAtas, BatasBawah, Tengah : Integer
Ketemu : boolean
ALGORITMA
Input (cari) { meminta nilai data yang akan dicari}
BatasAtas ← 1 { indeks array dimulai dari 1 }
BatasBawah ← N
Ketemu ← False
While (BatasAtas < BatasBawah) and (not ketemu) do
    Tengah ← (BatasAtas + BatasBawah) div 2
    If A [Tengah] = cari then
        Ketemu ← true
    Else
        If ( A [Tengah] < cari ) then { cari di bagian kanan }
            BatasAtas ← Tengah + 1
        Else
            BatasBawah ← Tengah - 1 { cari di bagian kiri }
        Endif
    Endif
EndWhile
If (ketemu) then
    Output ( 'Data berada di index nomor', Tengah )
Else
    Output ( 'Data tidak ditemukan' )
Endif
    
```

2	35	7	4	6	1	9	4	23	25
---	----	---	---	---	---	---	---	----	----

Gambar 3 . 1 Komplikasi Algoritma Binary Search

4. Hasil dan Pembahasan

4.1 Hasil Pengujian Pencarian

Dari langkah langkah algoritma yang sudah dijelaskan sebelumnya kemudian akan dilakukan pengujian, untuk melihat hasil *run time* pencarian data mahasiswa Program Studi Manajemen di Universitas Pelita Bangsa.

Pengujian pada aplikasi dilakukan sebanyak 10 kali pengujian pencarian data mahasiswa dan digunakan metode pengujian *black box* dijelaskan sebagai berikut :



Gambar 4.1 Hasil Pengujian

Hasil Pengujian, *input* nim dari data mahasiswa. Nim yang dicari “111811520” dan ditemukan atas nama “EVA MARGIYANI”, *run time* pencarian data mahasiswa 0.044 ms. Berikut tampilan dari hasil pencarian data mahasiswa.

Tabel 4.1 Hasil Pengujian Sepuluh Kali Percobaan



Pengujian	NIM yang Dicari	Hasil yang Dicari	Run Time
Pengujian 1	192010077	MIA ADITYA	0.041 ms
Pengujian 2	111810183	YULINAR	0.042 ms
Pengujian 3	111810377	WIDIA SAFITRI	0.024 ms
Pengujian 4	111810640	FAUZIAH SONIAH	0.038 ms
Pengujian 5	111810735	MUHAMMAD SAFE'I	0.035 ms
Pengujian 6	111810794	FAJAR DWI PRASETYO	0.035 ms
Pengujian 7	111810846	RENDI GAOL	0.022 ms

Pengujian 8	111810955	KURNIAWATI	0.044 ms
Pengujian 9	111811242	LUKMAN TRI LEGOWO	0.03 ms
Pengujian 10	111811520	EVA MARGIYANTI	1.44

4.2. Skenario Black Box Testing

Dalam tahap penelitian *Black Box Testing* ini langkah pertama *user* menuliskan nim mahasiswa yang ingin di cari pada sistem android tersebut , lalu klik tombol telusur maka proses pencarian berjalan, apabila NIM mahasiswa tersebut sesuai yang ada didalam *datbase* maka dinyatakan NIM tersebut valid maka akan menampilkan pemberitahuan hasil pencarian dan ketika nim tersebut sudah menampilkan NIM dan nama mahasiswa maka sistem juga akan menampilkan *run time* waktu eksekusi pencarian data mahasiswa program studi manajemen pada universitas pelita bangsa seperti table di bawah ini :

Tabel 4.2 Hasil Pengujian Black Box Testing

Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
Menulis NIM mahasiswa dan tekan tombol telusur	Mencari data mahasiswa yang ingin dicari 	Sesuai	Valid
Menampilkan hasil pencarian	Menampilkan hasil pencarian data mahasiswa yang dicari dan <i>run time</i> 	Sesuai	Valid

4.3. Kompleksitas Waktu Pencarian

Cari data 100

Nama : Rsila Astri Ditantri

Nim : 111810099

Lefth	mid	Right
0	999	1999
Lefth	mid	Right
0	299	499
Lefth	mid	Right
0	124	299
Lefth	mid	Right
0	62	124
Lefth	mid	Right

62				93					124
Lefth			mid			Right			
93				100					124

Stop sampai perhitungan 100. Setiap perhitungan dihitung 0,01 diatas terdapat 6 kali hitungan berarti 6 x 0,01 = 0,06

Cari data 500

Nama : Nasrul Aziz

Nim 111810501

Lefth				mid					Right
0				999					1999
Lefth									Right
0				459					999

Stop di angka 499 = 500 Setiap perhitungan dihitung 0,01 di atas terdapat 2 kali hitungan berarti 2 x 0,02 = 0,02

Cari data 1000

Nama : Riza Melya

Nim 111811003

Lefth				mid					Right
0				999					1999

Stop di angka 999 = 1000 Setiap perhitungan dihitung 0,01 di atas terdapat 1 kali hitungan berarti 1 x 0,02 = 0,01.

Cari data 1500

Nama : Andri

Nim : 111811509

Lefth				mid					Right
0				999					1999
Lefth									Right
0				1459					999

Stop di angka 1499 = 1500 Setiap perhitungan dihitung 0,01 di atas terdapat 2 kali hitungan berarti 2 x 0,01 = 0,02.

Waktu rata – rata = jumlah hasil pengujian di bagi nilai hasil penguji = 0,01 + 0,01 + 0,02 + 0,02 + 0,06 = 1 : 4 = 0,025 ms

5. Penutup

5.1 Kesimpulan

Berdasarkan uraian yang telah penulis jabarkan dapat diambil kesimpulan yaitu algoritma *searching* merupakan algoritma yang penting dalam pengelolaan sistem manajemen database. Data yang besar (*Big Data*) perlu diolah untuk mempermudah dalam pencarian data. Analisis kecepatan rata - rata waktu pencarian dan ketepatan algoritma *binary search* diketahui dengan perhitungan kompleksitas waktu dan kompleksitas algoritma menggunakan *binary search* diketahui dengan perhitungan kompleksitas waktu hasil dari penelitian ini diketahui analisis kecepatan rata-rata waktu 0,025 ms dalam jumlah data mahasiswa untuk membantu proses analisis sebanyak 2000 data berupa nim dan nama mahasiswa. Pengujian dalam penelitian ini menggunakan analisa kompleksitas waktu pencarian, analisa kompleksitas dengan notasi linear $O(n)$, dan *black box testing*. *Black box testing* hasil pengujian sesuai dan valid.

Daftar Pustaka

Kusnendi, Informasi and P. ISSN, Vol. 1-12, thn 2012, 2012.

- Siswanto, Pengantar Sistem Informasi Geografik, Vol 8 no. March, 2015.
- Dewi Martina, Konsep Dasar Sistem Informasi, Vol. I, pp. 5–10, 2012.
- D.A. Lantana, Metode Binary Search, Vol.2 no.6 pp. 1–7, 2012.
- G. G. Maulana, Pembelajaran Dasar Algoritma Dan Pemrograman Menggunakan El-Goritma Berbasis Web, *J. Tek. Mesin*, Vol. 6, no. 2, p. 8, 2017.
- D. Kusumasari and O. Rafizan, Studi Implementasi Sistem Big Data Untuk Mendukung Kebijakan Komunikasi Dan Informatika, *Masy. Telemat. Dan Inf. J. Penelit. Teknol. Inf. dan Komun.*, Vol. 8, no. 2, p. 81, 2018.
- M. F. Rohim, R. Hapsari, Y. Religia, and D. Prasetyo, “Analisis Algoritma Sequential Search Dan Binary Search Pada Big Data.vol 9 thn 2015”
- N. Rusmania, Tren Penelitian Reproduksi Gambar Stereo Gambar Ruang, *Nhk Technical Research*, Vol. 151, pp. 10–17, 2015.
- G. Syahputra and B. Sinurat, “Implementasi Teknik Binary Search Pada Kamus Indonesia - Batak Toba,” *J. Informatics Pelita Nusant.*, vol. 1, no. 1, pp. 28–37, 2016.
- B. Sulistio *et al.*, Aplikasi Kamus Bahasa Taliabu Berbasis Android Dengan Android Based Dictionary Taliabu Language Using Binary Search, Vol. 2, no. 2, pp. 67–72, 2019
- D. M. Andayani, M. Yuliana, T. B. Santoso, L. Sinyal, P. Elektronika, and N. Surabaya, “Penggunaan Metode Binary Search Pada Translator Bahasa Indonesia – Bahasa Jawa,” pp. 1–8.
- S. Rahman, “Perancangan Aplikasi Remote Desktop Viewer Menggunakan Metode Pencarian Biner (Binary Search). Vol 2, no 3 pp5-10”
- Bayu Sulistio1, “Dicetak pada tanggal 2018-09-04 Id Doc: , Vol 8 pp. 13–38, 2018.
- Panny Agustia Rahayuningsih P. Pengembangan and A. Dan, Vol 8, Bab I,” pp. 1–31, 2017
- Ridwan H.. Dari, A. Pencarian, and B. P. A. E-acesia, “Hashtable Sebagai Alternatif Dari Algoritma Pencarian Biner Pada Aplikasi E-Acesia,” *J. Inform.*, vol. 8, no. 2, pp. 943–952, 2014.
- C. Pralystia, Tinjauan Sistem Informasi, Vol 8, pp. 11–59, 2009.
- J. O. Papilaya and N. Huliselan, “Identifikasi Gaya Belajar Mahasiswa,” *J. Psikol. Undip*, vol. 15, no. 1, p. 56, 2016.
- J. I. Maanari, R. Sengkey, I. H. F. Wowor, and Y.D.Y. Rindengan, Perancangan Basis Data Perusahaan dengan Menggunakan Oracle, *E-Journal Tek. Elektro dan Komputer*, Vol. 2., No. 2, 201