ANALISIS WAKTU PROSES PENCARIAN NOMOR HADIST YANG HILANG DARI KOLEKSI HADIST MENGGUNAKAN TIGA ALGORITMA

OLEH:

REZHA RAMADHANA

JURUSAN ILMU KOMPUTER

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS LAMPUNG

2019

ABSTRACT

ANALYSIS OF HADIST NUMBER SEARCHING PROCESSING MISSING FROM HADIST COLLECTION USING THREE ALGORITHM

BY

REZHA RAMADHANA

Islam has several sources for making decisions based on the teachings of Islamic law, one of which is the Hadith. Hadiths are often used to strengthen the laws established by the Koran and to establish new laws not stipulated by the Koran. As technology advances, many hadith search information systems have been created. In the hadith search system that has been made before, there are some collections of hadith books that are incomplete. In this study the authors aim to find the number of lost hadith from several collections of hadith books that have been collected. The data used are 4292 Abu Daud Hadith, 14509 Ahmad Hadith, Bukhari 7008 Hadith, and Malik Hadith 1594. There are three algorithms used to find the missing hadith numbers. Hadith collection data that have been collected are tested using three algorithms. Then the results of the test are analyzed and the most efficient algorithm is obtained based on the processing time.

Keywords: Hadith, Abu Daud, Ahmad, Bukhari, Malik, Algorithms, processing time

ABSTRAK

ANALISIS WAKTU PROSES PENCARIAN NOMOR HADIST YANG HILANG DARI KOLEKSI HADIST MENGGUNAKAN TIGA ALGORITMA

OLEH

REZHA RAMADHANA

Islam memiliki beberapa sumber untuk mengambil keputusan bedasarkan ajaran hukum Islam, salah satu sumbernya adalah Hadist. Hadist sering kali digunakan untuk menguatkan hukum yang ditetapkan oleh Al-Quran serta menetapkan hukum baru yang tidak ditetapkan oleh Al-Quran. Seiring dengan kemajuan teknologi, banyak sistem informasi pencarian hadist yang telah dibuat. Pada sistem pencarian hadist yang telah dibuat sebelumnya, ada beberapa koleksi kitab hadist yang kurang lengkap. Pada penelitian kali ini penulis bertujuan untuk mencari nomor hadist yang hilang dari beberapa koleksi kitab hadist yang telah dikumpulkan. Data yang digunakan adalah Hadist Abu Daud sebanyak 4292, Hadist Ahmad sebanyak 14509, Hadist Bukhari sebanyak 7008, dan Hadist Malik sebanyak 1594. Terdapat tiga algoritma yang digunakan untuk mencari nomor hadist yang hilang. Data koleksi hadist yang telah dikumpulkan dilakukan pengujian menggunakan tiga algortima. Kemudian hasil dari pengujian dianalisa dan didapatkan algoritma yang paling efisien bedasarkan waktu proses.

Kata kunci: Hadist, Abu Daud, Ahmad, Bukhari, Malik, Algoritma, waktu proses

ANALISIS WAKTU PROSES PENCARIAN NOMOR HADIST YANG HILANG DARI KOLEKSI HADIST MENGGUNAKAN TIGA ALGORITMA

Oleh

Rezha Ramadhana

Skripsi

Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA KOMPUTER

Pada

Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS LAMPUNG

BANDAR LAMPUNG

2019

Judul Skripsi	:	**ANALISIS WAKTU PROSES PENCARIAN NOMOR HADIST YANG HILANG DARI KOLEKSI HADIST MENGGUNAKAN TIGA ALGORITMA**
Nama Mahasiswa	:	Rezha Ramadhana
Nomor Pokok Mahasiswa	:	1347051013
Jurusan	:	Ilmu Komputer
Fakultas	:	Matematika dan Ilmu Pengetahuan Alam

MENYETUJUI

1. Komisi Pembimbing

Didik Kurniawan, S.Si., M.T.
NIP 19800419 200501 1 004

2. Ketua Jurusan Ilmu Komputer

Dr. Ir. Kurnia Muludi, M.S.Sc.
NIP 19640616 198902 1 001

MENGESAHKAN

1. Tim Penguji

Ketua : **Didik Kurniawan, S.Si., M.T.** .....

Penguji
Bukan Pembimbing : **Dr. Ir. Kurnia Muludi, M.S.Sc.** .....

Penguji
Bukan Pembimbing : **Bambang Hermanto, S.Kom., M.Cs.** .....

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Dr. Suratman, M.Sc.
NIP. 19640604 199003 1 002

Tanggal Lulus Ujian Skripsi : **3 Desember 2019**

PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi saya yang berjudul "ANALISIS WAKTU PROSES PENCARIAN NOMOR HADIST YANG HILANG DARI KOLEKSI HADIST MENGGUNAKAN TIGA ALGORITMA" merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang saya terima.

Bandar Lampung, 15 Desember 2019

REZHA RAMADHANA.
NPM. 1347051013

RIWAYAT HIDUP

Penulis dilahirkan pada tanggal 27 Februari 1995 di Braja Sakti Lampung Timur, Lampung sebagai anak pertama dari tiga bersaudara dengan Ayah yang bernama Rudiwanto dan Ibu bernama Sumarini. Penulis menyelesaikan pendidikan dasar di SDN 1 Labuhan Ratu Lampung Timur, selesai pada tahun 2007. Kemudian penulis melanjut ke pendidikan menengah pertama di SMPN 1 Way Jepara Lampung Timur dan diselesaikan penulis pada tahun 2010. Kemudian melanjutkan ke pendidikan menengah atas di SMK TELKOM Purwokerto dan diselesaikan pada tahun 2013.

Pada tahun 2013 penulis terdaftar sebagai mahasiswa di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung melalui jalur Mandiri. Selama dalam masa perkuliahan, penulis mengikuti organisasi internal Jurusan yaitu Himakom (Himpunan Mahasiswa Jurusan Ilmu Komputer) periode 2013/2014. Selama menjadi mahasiswa beberapa kegiatan yang dilakukan penulis antara lain pada bulan Februari 2016 penulis melaksanakan Kerja Praktik (KP) di KPU Bandar Lampung, dan pada bulan Juli 2016 Penulis melaksanakan Kuliah Kerja Nyata (KKN) di Lampung Tengah.

PERSEMBAHAN

Segala puji syukur atas berkah dan rahmat dari Allah Subhanalllah Wata'ala, saya persembahkan skripsi ini untuk orang-orang yang selalu saya harapkan cinta dan kasih sayangnya.

Teruntuk Ibu dan Ayah yang tidak pernah memutus doa untuk anak-anaknya, terima kasih untuk dukungan dan tuntunan kalian, semoga kelak saya menjadi anak yang berguna dan membuat bangga kalian.

Teruntuk keluarga dan sanak famili, terima kasih atas teguran, motivasi dan dukungan dari kalian.

Teruntuk teman-teman tercinta, terima kasih untuk selalu ada dan mendukung di saat senang dan sulit.

Almamater Tercinta,

UNIVERSITAS LAMPUNG

MOTTO

"Dan Allah selalu bersama orang – orang yang sabar"
(Al-Anfal ayat 66)

"Sesungguhnya sesudah kesulitan itu ada kemudahan, sesungguhnya sesudah kesulitan itu ada kemudahan"
(Asy Syarh ayat 5-6)

"Sedikit lebih beda lebih baik daripada sedikit lebih baik"
(Pandji Pragiwaksono)

SANWACANA

Assallamu'alaikum Warahmatullahi Wabarakatu..

Puji syukur kehadiran Allah Subhanallahu Wata'ala karena atas berkah dan rahmat-Nya penulis dapat menyelesaikan skripsi ini yang berjudul "Analisis Waktu Proses Pencarian Nomor Hadist Yang Hilang Dari Koleksi Hadist Menggunakan Tiga Algoritma". Skripsi ini merupakan salah satu syarat untuk memperoleh gelar Sarjana Komputer di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.

Dalam pelaksanaan dan penyusunan skripsi ini Penulis sangat berterima kasih dan memberikan penghargaan yang sedalam-dalamnya kepada seluruh pihak yang membantu menyelesaikan skripsi ini. Penulis ingin mengucapkan terima kasih dengan setulus hati terutama kepada:

1. Kedua Orang tua tercinta, yaitu Ayah dan Ibu, serta Adik. serta saudara-saudaraku yang selalu saya sayangi dan kasihi yang selalu memberikan dukungan, masukan, motivasi, dan do'anya yang tak terhingga.
2. Bapak Drs. Suratman, M.Sc. selaku dekan FMIPA Universitas Lampung.
3. Bapak Dr. Ir. Kurnia Muludi, M.S.Sc. selaku ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung.

4. Bapak Didik Kurniawan, S.Si., MT. sebagai pembimbing utama skripsi atas kesediaannya, kesabaran dan keikhlasannya untuk memberikan dukungan, bimbingan, nasihat, saran, dan kritik dalam proses penyelesaian skripsi ini.

5. Bapak Dr. Ir. Kurnia Muludi, M.S.Sc. selaku pembahas skripsi, yang telah memberikan saran dan masukan guna penyempurnaan dalam penulisan skripsi ini.

6. Bapak Bambang Hermanto, S.Kom.,M.Cs. selaku pembahas skripsi, yang telah memberikan saran dan masukan guna penyempurnaan dalam penulisan skripsi ini.

7. Ibu Ade Nora Maela, Mas Nofal yang telah membantu memudahkan segala urusan administrasi penulis di Jurusan Ilmu Komputer.

8. Seluruh keluarga dan saudaraku yang telah memberikan dukungan selama proses perkuliahan yang tidak bisa disebutkan satu persatu.

9. Teman-teman tercinta Esa, Yudi, Tomi, Firman, Mamang, Arif, Radit, terima kasih banyak untuk selalu ada baik dalam keadaan suka maupun duka.

10. Rekan-rekan Ilmu Komputer 2013 yang tidak bisa disebutkan satu persatu, terima kasih untuk segala dukungan, bantuan, serta kebersamaannya selama ini.

11. Semua pihak yang secara langsung maupun tidak langsung yang telah membantu dalam penyelesaian skripsi ini.

Penulis telah berusaha semaksimal mungkin dalam penulisan skripsi ini untuk mencapai suatu kelengkapan dan kesempurnaan. Penulis juga mengharapkan saran dan kritik yang bersifat membangun dari semua pihak. Akhirnya dengan segala kerendahan hati penulis berharap skripsi ini dapat memberikan manfaat, baik kepada penulis khususnya maupun kepada pembaca pada umumnya.

Bandar Lampung, Desember 2019

Penulis,

Rezha Ramadhana

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Gambar Halaman

DAFTAR TABEL

Tabel Halaman

BAB I

PENDAHULUAN

1.1 Latar Belakang

Algoritma merupakan kumpulan perintah untuk menyelesaikan suatu masalah. Perintah-perintah ini dapat diterjemahkan seacara bertahap dari awal hingga akhir. Salah satu fungsi dari algoritma adalah dapat digunakan untuk pencarian data, dengan algoritma data yang dibutuhkan dapat dicari sesuai dengan masukan yang diberikan. Pada penelitian ini akan dilakukan analisis algoritma pencarian nomor yang hilang dari beberapa kitab hadist, data hadist yang akan digunakan telah ditentukan sebelumnya, yaitu hadist Bukhari, hadist Malik, hadist Ahmad, dan hadist Abu Daud. Terdapat tiga algoritma yang digunakan untuk mencari nomor yang hilang dari beberapa koleksi hadist di atas, kemudian dari ketiga algortima tersebut akan dilakukan analisis untuk menentukan manakah algoritma yang paling efisien bedasarkan waku prosesnya.

1.2 Rumusan Masalah

Masalah yang akan diabahas dalam penelitian ini adalah menganalisis manakah algoritma yang paling efisien bedasarkan waktu proses.

1.3 Batasan Masalah

Adapun batasan masalah dalam penelitian ini:

1. Menguji algoritma yang dibuat dan membandingkan algoritma bedasarkan waktu proses (*running time)*.
2. Menemukan nomor hadist yang hilang dari beberapa koleksi kitab hadist.

1.4 Tujuan

Tujuan dari penelitian ini adalah menemukan nomor hadist yang hilang dan menganalisis hasil *running-time* dari algoritma yang dibuat.

1.5 Manfaat

Manfaat dari penelitian ini adalah menemukan nomor hadist yang hilang dari beberapa koleksi kitab hadist dan menentukan manakah algoritma yang paling efisien bedasarkan waktu proses.

BAB II

TINJAUAN PUSTAKA

2.1 Algoritma

Algoritma merupakan prosedur atau urutan langkah-langkah untuk memecahkan suatu masalah. Dalam dunia pemrograman, algoritma merupakan sebuah skema dimana langkah-langkah suatu proses dikerjakan secara berurutan sesuai dengan urutan langkah kerja algoritma. Setiap langkah harus didefinisikan dengan tepat dan tidak ambigu, setiap langkah dalam algortima juga harus sederhana sehingga dapat dikerjakan dalam sejumlah waktu yang masuk akal. Sebuah algortima dapat memiliki nol atau lebih masukan, dan mempunyai nol atau lebih keluaran. Setelah mengerjakan sejumlah langkah terbatas, algoritma harus berhenti dengan kata lain jika suatu program yang merupakan implementasi dari suatu algoritma tidak pernah berhenti maka dapat mengindikasikan bahwa program tersebut berisi algoritma yang salah (Munir, 2011).

Algoritma berisi deskripsi langkah-langkah penyelesaian masalah. Langkah-langkah penyelesaian tersebut dapat dituliskan dalam notasi algoritmik. Tidak seperti pada notasi bahasa pemrograman, tidak ada notasi yang standar atau aturan baku unuk menuliskan algortima, setiap orang dapat mendefinisikan notasi algoritmiknya sendiri. Notasi algoritmik yang baik adalah notasi yang mudah dibaca dan mudah pula ditranslasikan ke dalam notasi Bahasa pemrograman. Dengan kata lain program komputer adalah implementasi algoritma dalam notasi bahasa pemrograman tertentu (Munir, 2011).

2.2 Kompleksitas Algoritma

Secara senderhana algoritma didefinisikan sebagai prosedur selangkah demi selangkah untuk mencari solusi suatu masalah dalam waktu yang terbatas. Umumnya algoritma mentransformasikan obyek masukan menjadi obyek keluaran dengan waktu eksekusi yang merupakan fungsi dari objek masukan. Dalam arti makin besar obyek masukan makin lama waktu yang dibutuhkan untuk menghasilkan obyek keluaran (Subandijo, 2011).

Komplesitas algoritma diukur bedasarkan kinerjanya dengan menghitung waktu eksekusi suatu algoritma. Waktu eksekusi algoritma dapat diklasifikasikan menjadi 3 kelompok besar, yaitu *best-case* (kasus terbaik), *average-case* (kasus rerata) dan *worst-case* (kasus terjelek). Pada pemrograman yang dimaksud dengan kasus terbaik, kasus terjelek dan kasus rerata suatu algoritma adalah besar kecilnya atau banyak sedikitnya sumber-sumber yang digunakan oleh suatu algoritma. Makin sedikit makin baik,

makin banyak makin jelek. Biasanya sumber-sumber yang paling dipertimbangkan tak hanya waktu eksekusi tetapi bisa juga besar memori, catu-daya dan sumber-sumber lain (Goldreich, 2008).

2.3 Notasi O-Besar

Notasi O-Besar adalah cara yang digunakan untuk menguraikan laju pertumbuhan suatu fungsi yang tidak lain adalah *time complexity* suatu algoritma. Notasi O-Besar pertama kali dikenalkan oleh pakar teori bilangan Paul Bachman pada tahun 1894 dalam bukunya *Analytische Zahlentheorie ("analytic number theory")* volume dan untuk menyatakan *asympotic upper bounds* suatu fungsi. Volume pertamanya diterbitkan tahun 1892 tetapi belum memuat notasi O. Notasi ini dipopulerkan pada teori bilangan oleh Edmund Landau yang kemudian kerap disebut simbol Landau. Pada ilmu komputer notasi O-Besar diklaim sudah dikenal sejak tahun 1927 yang kemudian dipopulerkan kembali penggunaannya oleh Knuth (1976, 1998, 1999) yang juga menggenalkan kembali notasi Omega dan Theta. Notasi O-Besar, dibaca sebagai "*order of*", aslinya menggunakan simbol *omikron* besar. Saat ini digunakan huruf besar Latin O yang kelihatannya mirip dengan *omikron* (Subandijo, 2011).

Misalkan telah di dapatkan pada algoritma bilangan genap, bahwa :

$\mathbf{T_{max}(n) = 1}$

- O (Big Oh)

$T(n) \in O(g(n))$ jika $T(n) \leq cg(n)$; untuk $n \geq n_0$. Dimana nilai konstanta (c) dan n_0 (batas n) nilainya bebas ditentukan.

$1 \leq n$ (untuk semua $n \geq 1$)

$c = 1$; $n_0 = 1$

- Ω (Big Omega)

$T(n) \in \Omega(g(n))$ jika $T(n) \geq cg(n)$; untuk $n \geq n_0$. Dimana nilai c (konstanta) dan n_0 (batas n) nilainya bebas ditentukan.

$1 \geq n$ (untuk semua $n \geq 0$)

$c = 1$; $n_0 = 0$

- Θ (Big Theta)

$T(n) \in \Theta(g\{n))$ jika $c_2g(n) \leq T(n) \leq c_1g(n)$; untuk $n \geq n_0$. Dimana c (konstanta) dan n_0 (batas n) nilainya bebas ditentukan.

Batas atas :

$1 \leq n$ (untuk semua $n \geq 1$)

Batas Bawah :

$1 \geq n$ (untuk semua $n \geq 0$)

$c_1 = 1$; $c_2 = 1$; $n_0 = 0$

$T_{min}(n) = n + 1 \approx n$

- O (Big Oh)

$T(n) \in O(g(n))$ jika $T(n) \leq cg(n)$; untuk $n \geq n_0$. Dimana nilai konstanta (c) dan n_0 (batas n) nilainya bebas ditentukan.

$n + 1 \leq n + n$ (untuk semua $n \geq 1$)

$n + 1 \leq 2n$ (untuk semua $n \geq 1$)

$c = 2$; $n_0 = 1$

- **Ω** (Big Omega)

$T(n) \in \Omega(g(n))$ jika $T(n) \geq cg(n)$; untuk $n \geq n_0$. Dimana nilai c (konstanta) dan n_0 (batas n) nilainya bebas ditentukan.

$n + 1 \geq n$ (untuk semua $n \geq 0$)

$c = 1$; $n_0 = 0$

- Θ (Big Theta)

$T(n) \in \Theta(g\{n))$ jika $c_2g(n) \leq T(n) \leq c_1g(n)$; untuk $n \geq n_0$. Dimana c (konstanta) dan n_0 (batas n) nilainya bebas ditentukan.

Batas atas :

$n + 1 \leq 2n$ (untuk semua $n \geq 1$)

Batas Bawah :

$n +1 \geq n$ (untuk semua $n \geq 0$)

$c_1 = 2$; $c_2 = 1$; $n_0 = 1$

$T_{avg}(n) = 1 + (n+1) / n$

- O (Big Oh)

$T(n) \in O(g(n))$ jika $T(n) \leq cg(n)$; untuk $n \geq n_0$. Dimana nilai konstanta (c) dan n_0 (batas n) nilainya bebas ditentukan.

$1 + (n + 1) / n \leq n + n$ (untuk semua $n \geq 1$)

$(n + 2) / n \leq 2n$ (untuk semua $n \geq 2$)

$c = 2$; $n_0 = 2$

- Ω (Big Omega)

$T(n) \in \Omega(g(n))$ jika $T(n) \geq cg(n)$; untuk $n \geq n_0$. Dimana nilai c (konstanta) dan n_0 (batas n) nilainya bebas ditentukan.

$1 + (n + 1) / n \geq n$ (untuk semua $n \geq 1$)

$(n + 2) / n \geq n$

$c = 1$; $n_0 = 1$

- Θ (Big Theta)

$T(n) \in \Theta(g\{n))$ jika $c_2 g(n) \leq T(n) \leq c_1 g(n)$; untuk $n \geq n_0$. Dimana c (konstanta) dan n_0 (batas n) nilainya bebas ditentukan.

Batas atas :

$1 + (n + 1) \leq 2n$ (untuk semua $n \geq 2$)

$(n + 2) / n \leq 2n$

Batas Bawah :

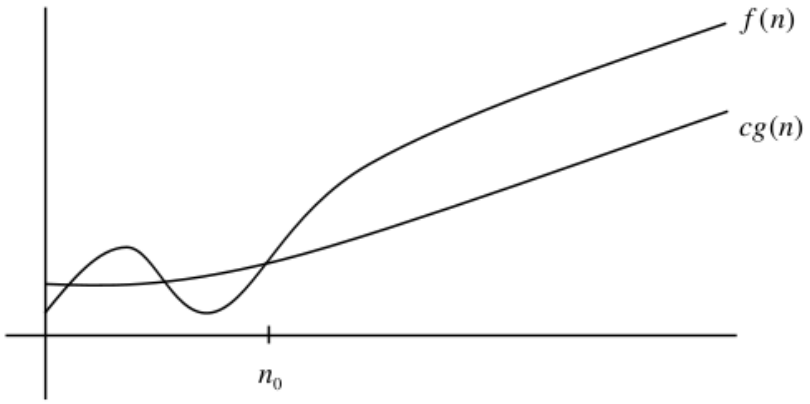
$1 + (n + 1) \geq n$ (untuk semua $n \geq 1$)

$(n + 2) / n \geq n$

$c_1 = 2$; $c_2 = 1$; $n_0 = 2$

Dalam matematika dan ilmu komputer, notasi O-Besar juga dikenal dengan notassi *Omikron* besar, notasi Landau, notasi Bachma-Landau dan notasi asimpotik. Bersama dengan notasi yang berkaitan seperti Omega besar, notasi Theta besar dan notasi o kecil, notasi O-Besar digunakan untuk menguraikan prilaku batas suatu fungsi jika argumen fungsi bergerak menuju nilai tak terbatas. Notasi O-Besar mengklasifikasikan fungsi bedasarkan kecepatan pertumbuhan argumennya, yaitu fungsi berbeda dengan kecepatan pertumbuhan yang sama akan disajikan dengan notasi O yang sama. Uraian fungsi yang berkaitan dengan notasi O-Besar biasanya hanya menyajikan batas atas kecepatan pertumbuhan fungsi (Higham, 1998).

Jika diberikan 2 buah fungsi *f* dan *g* yang keduanya merupakan fungsi pada bilangan real. $f(n) = \Omega(g(n))$ dikatakan $f(n)$ Big Omega dari $g(n)$, jika terdapat dua buah bilangan real positif C dan $n_{0,}$ untuk semua $n \geq n_0$ berlaku $Cg(n) \leq f(n)$. Gambar geometriknya :



Gambar 2.1 Geometrik Big-O dan Big-Omega

Hubungan dari Big-O dan Big-Omega diatas seperti berikut:

$f(n) = O(g(n))$ jika hanya jika $g(n) = \Omega(f(n))$.

2.4 Efisiensi Algoritna

Ada sejumlah terminologi yang terlebih dahulu perlu dipahami untuk mengukur efisiensi algoritma. Biasanya efisiensi algoritma dinyatakan sebagai berapa lama ia dieksekusi bedasarkan banyak data masukan yang dinyatakan dalam *n*. Ada dua pendekatan utama yang dapat digunakan untuk menghitung efesiensi yaitu studi eksprimental dan analisis teoritis (Arora and Barak, 2009).

Studi eksprimental dilakukan dengan menulis program untuk mengimplementasikan algoritma berdasarkan *hardware* yang berbeda, tetapi metode ini jarang digunakan dengan alasan proses yang dilakukan menggunkan dua *hardware* yang berbeda. Oleh karena itu alternatifnya adalah analisis teoritis. Pendekatan ini tidak menggunakan implementasi dengan cara menulis program tetapi mnggunakan deskripsi algoritma yang dikenal dengan nama *pseudocode* dengan karakteristik utama waktu eksekusi sebagai fungsi dari ukuran obyek masukan *n* sehingga memungkinkan memasukkan semua obyek data yang mungkin. Pendekatan ini memungkinkan mengestimasi kecepatan algoritma yang tidak tergantung pada lingkungan pengembangan

program seperti *hardware* dan *software*. Hal ini disebabkan karena algoritma adalah *platform independent*, dalam arti dapat diimplementasikan dalam sembarang bahasa pemrograman pada sembarang komputer yang menggunakan sembarang sistem operasi.

Operasi dasar yang dilakukan oleh algortima yang dapat diidentifikasi pada *pseudocode* disebut operasi *primitive*. Contoh operasi *primitive* yang banyak dikenal adalah evaluasi ekspresi, pengindeksan pada *array*, pemanggilan metode dan sebagainya. Sebagian besar dari mereka tidak tergantung pada bahasa pemrograman dan diasumsikan memerlukan waktu akses konstan pada memori. Dengan mengamati *pseudocode* banyak operasi *primitive* maksimum suatu algoritma sebagai fungsi dari masukan dapat dihitung.

Sebagain contoh:

Algoritma *arrayMax*(A,n)	# operasi promitif
curMax ← *A[0]*	*1*
for I ← *1 to n-1 do*	*n-1*
if A[i] > curMax then	*n-1*
curMax ← *A[i]*	*n-1*
{ increment counter i }	*n-1*
return curMax	*1*

Algoritma di atas mengeksekusi 4n-2 operasi *primitive*. Jika didefinisikan *a* dan *b* sebagai waktu tercepat dan waktu terlambat berturut-turut yang dilakukan oleh operasi *primitive* dan T(n) sebagai *worst-case time* dari *arrayMax* maka didapat.

$$a\,(4n-2) \leq T(n) \leq b\,(4n-2)$$

Ini berarti bahwa eksekusi T(n) dibatasi oleh dua fungsi linear n. perubahan *hardware* dan *software* hanya akan mempengaruhi T(n) dengan factor konstan tetapi tidak akan mempengaruhi laju pertumbuhan T(n) karena laju pertumbuhan linear waktu eksekusi T(n) merupakan karakteristik *intrinsic* dari *arrayMax*.

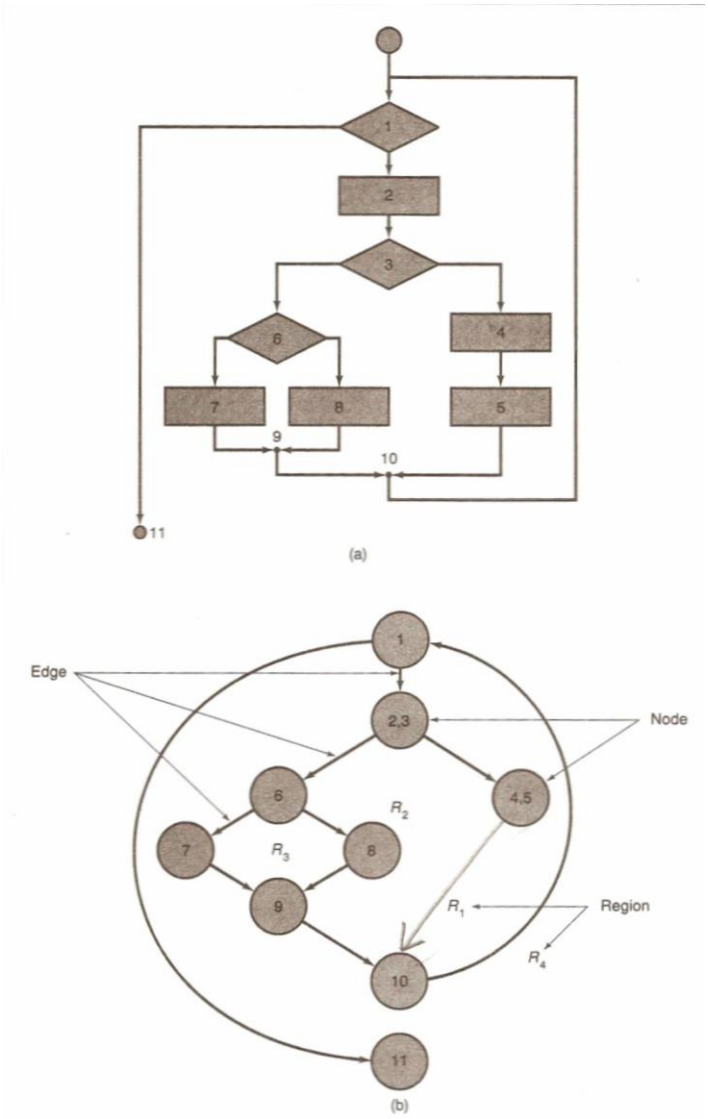
Pendekatan lain juga dapat digunakan untuk menentukan nilai *order* algoritma O-Besar adalah berasumsi bahwa *order* algoritma adalah O(1), algoritma tidak melakukan apapun dan lansung berakhir apapun bentuk data masukannya. Kemudian cari bagian kode yang diduga mempunyai *order* tertinggi. Mulai dari sini, penyelidikan efesiensi algoritma dimulai dari luar ke dalam. Efesiensi total adalah hasil kali efesiensi di setiap lapisan kode.

2.5 *Cyclomatic Complexity*

Cyclomatic Complexity merupakan *software* matrik yang menyediakan ukuran kuantitatif dari kompleksitas logikal suatu program. Ketika digunakan dalam konteks metode uji coba berbasis alur, nilai yang dikomputasi untuk kompleksitas *cyclomatic* mendefinisikan jumlah *independent path* dalam

himpunan basis suatu program dan menyediakan batas atas untuk spejumlah uji coba yang harus dilakukan untuk memastikan bahwa seluruh perintah telah dieksekusi sedikitnya satu kali (Liana, 2009).

Independent path adalah alur manupun dalam program yang memperkenalkan sedikitnya satu kumpulan perintah pemrosesan atau kondisi baru. Contoh *independent path* dari Gambar 2.2.



Gambar 2.2 *independent path* (Liana, 2009)

2.6 Waktu Proses (*Running-Time*)

Waktu proses program (*running time*) adalah besaran waktu yang dibutuhkan program untuk melakukan eksekusi. Besaran ini menunjukkan seberapa lama program menyelesaikan masalah. Semakin besar nilai *running time* suatu program maka semakin tidak efektif.

Masalah yang menyangkut analisis algoritma sangat luas, untuk itu diperlukan pembatasan. Salah satu yang paling dominan adalah kompleksitas ruang, bentuk kompleksitas ruang ini adalah *running time* yaitu untuk menentukan waktu eksekusi suatu algoritma. *Running time* algoritma digunakan untuk mengetahui lama suatu program untuk menyelesaikan eksekusi. Bentuk *running time* ini adalah suatu fungsi dengan variabel n yang menunjukkan besaran waktu untuk jumlah data sebanyak n data (Setiawan, 2013).

2.7 Kompleksitas Waktu

Jumlah tahapan komputasi dihitung dari berapa kali suatu operasi dilaksanakan di dalam sebuah algorita sebagai fungsi ukuran masukan.

Kompleksitas waktu dibedakan menjadi 3 macam yaitu:

1.T(max) → kompleksitas waktu untuk kasus terburuk (*worst case*) atau kebutuhan waktu maksimum.

2.T(min) → kompleksitas waktu untuk kasus terbaik (*best case*) atau kebutuhan waktu minimum.

3.T(avg) → kompleksitas waktu untuk kasus rata-rata (*average case*) atau kebutuhan waktu rata-rata.

2.8 Penomoran Hadist

Penomoran hadits ini memang bisa dibilang sesuatu yang baru dalam keilmuan hadits. Hal itu tak lain sebenarnya untuk memudahkan dalam melacak atau juga untuk diingat. Masalah yang sering muncul terkait nomor hadits ini adalah ketidak samaan nomor hadits dalam satu judul kitab. Misalnya: Hadits Shahih Bukhari nomor 500, antar satu kitab dengan kitab lainnya bisa jadi berbeda. Biasanya karena percetakannya yang berbeda (Luthfi, 2017).

Hal ini dapat mengakibatkan nomor yang berbeda, karena standar tiap percetakan dalam pemberian nomor juga berbeda. Misalnya ada percetakan yang memberi nomor tiap bab. Ketika masuk bab berikutnya, nomornya kembali dari angka 1. Ada yang memberi nomornya berkesinambungan sampai akhir kitab. Bisa dikatakan penomoran hadits itu bukan tradisi dari ulama salaf. Bahkan Imam Bukhari sendiri juga tak menomori hadits di kitabnya. Meski bukan tradisi ulama salaf bukan berarti selalu jelek. Biasanya mereka menyebutkan hadits ini riwayat siapa di bab apa dari shahabat siapa. Penomoran hadits itu tak jauh beda dengan pemberian warna-warni yang berbeda dalam tulisan sebuah buku modern. Paling tidak, hal itu bisa kita simpulkan dari pernyataan Musthafa al-Bugha dalam mukaddimah tahqiq hadits Shahih Bukhari atau yang bernama *Minhat al-Bari fi Khidmat Shahih al-Bukhari.* Beliau menyebut:

> *Kitab ini banyak kita temukan belum ada seni percetakan modern, seperti antar satu bab dan bab lain ada pemisahnya, haditsnya juga belum ada nomornya atau awalan bab yang berbeda, sehingga menjadikan pembaca mudah untuk mencarinya kembali.* (Mushtafa al-Bugha, *Minhat al-Bari fi Khidmat Shahih al-Bukhari*)

Ada juga penomoran hadits yang dilakukan oleh Muhammad Fuad Abdul Baqi ini terinspirasi oleh salah seorang Orientalis. Orientalis itu bernama Arent J. Wensinck seorang Profesor bahasa Semit, termasuk bahasa Arab di Universitas Leiden, negeri Belanda. Dia membuat kamus untuk mempermudah mencari satu hadits di banyak kitab hadits. Karena sifatnya kamus, maka harus ringkas dan cepat untuk dilacak. Dapat disimpulkan penomoran hadist itu bukan seesuatu yang baku, tergantung siapa yang memberikan nomor hadist tersebut

BAB III

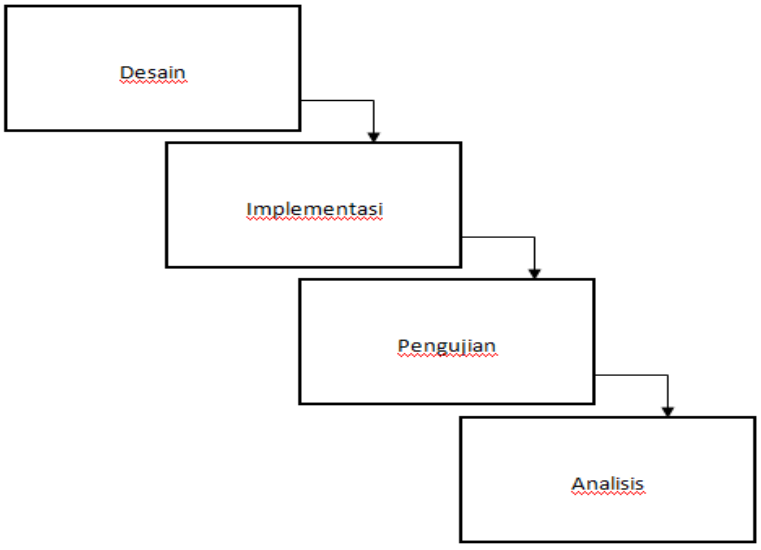
METODOLOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian

Penelitian ini dilakukan di Universitas Lampung Jurusan Ilmu Komputer Jl. Sumantri Brojonegoro No. 1 Gedung Meneng Bandar Lampung Telp. (0721) 704624, fax (0721) 704624. Waktu penelitian dilakukan pada Semester Ganjil Tahun Ajaran 2017-2018 dengan sumber data hadist didapat dari sistem hadist yang pernah dirancang sebelumnya.

3.2 Metode Penelitian

Penelitian ini dilakukan bedasarkan diagram alir metodologi penelitian sepeti berikut:



Gambar 3.1 Diagram Alir Metodologi Peneliitian

Penjelasan dari diagram alir metodologi penelitian pada Gambar 3.1 adalah sebagai berikut:

1. Desain

   Pada tahap desain dilakukan perancangan proses alur algoritma yang akan digunakan untuk mencari nomor yang hilang dari beberapa kitab hadist.

2. Implementasi

   Tahap ini merupakan tahap dimana menerapkan hasil rancangan desain dari algoritma yang sudah dibuat dalam bahasa pemrograman php.

3. Pengujian

   Pengujian pada algoritma yang telah diimplementasikan dilakukan untuk mengecek apakah algoritma berjalan dan menghasilkan *output* yang sesuai.

4. Analisis

   Setelah dilakukan pengujian, hasil dari pengujian kemudian dianalisa untuk menentukan manakah algoritma yang paling efisien bedasarkan waktu proses.

3.3 Metode Pengumpulan Data

Kegiatan yang merupakan suatu studi dengan teknik pengumpulan data dengan cara mempelajari literatur yang ada untuk mendukung landasan teoritis penulisan serta pengumpulan data dari penelitian-penelitian sebelumnya yang berkaitan dengan masalah yang akan dibahas sebagai pendukung penelitian. Studi literatur yang digunakan adalah buku-buku, jurnal, dan internet yang menyajikan informasi tentang analisis algoritma bedasarkan waktu proses (*running-time*). Data yang digunakan untuk penelitian ini sebagai berikut:

1. Hadist Abu Daud sebanyak 4292 dari total keseluruhan hadist 4590 yang dikumpulkan oleh Sunita Agustina.
2. Hadist Bukhari sebanyak 7008 dari total keseluruhan hadist 7008 yang dikumpulkan oleh Jaka PurnamaSidi.
3. Hadist Malik sebanyak 1594 dari total keseluruhan hadist 2286 yang dikumpulkan oleh Rifqi Ziyaddurohman.
4. Hadist Ahmad sebanyak 14509 dari total keseluruhan hadist 26363 yang dikumpulkan oleh Sunita Agustina.

3.4 Metode Analisis Algoritma

1. Mengimplementasikan algoritma yang sudah dibuat ke dalam sebuah program untuk melakukan pengujian.
2. Pengujian dilakukan terhadap setiap data Hadist menggunakan tiga algoritma yang sudah dibuat.
3. Ulangi pengujian data untuk mendapatkan waktu terbaik(*best case*), waktu terburuk(*worst case*), dan waktu rata-rata(*average case*).
4. Tuangkan hasil pengujian ke dalam bentuk tabel yang menunjukan hasil waktu proses pada setiap data hadist untuk setiap pengujian.
5. Bandingkan setiap rata-rata hasil pengujian dari ketiga algoritma menggunakan grafik atau *chart*.

BAB V

PENUTUP

5.1 Kesimpulan

Bedasarkan penelitian yang telah dilakukan diperoleh beberapa kesimpulan sebagain berikut:

1. Dari sistem pencarian Hadist yang telah dibuat sebelumnya terbukti bahwa masih ada beberapa data Hadist yang hilang.
2. Algoritma berhasil melakukan pencarian terhadap nomor Hadist yang hilang.
3. Algoritma menghasilkan waktu proses yang berbeda-beda terhadap hasil uji dari setiap Hadist.
4. Dari hasil uji coba yang dilakukan Algoritma 1 menjadi yang paling efisien bedasarkan waktu prosesnya.

5.2 Saran

Adapun saran yang diberikan sebagai berikut:

1. Algoritma yang dibuat hanya sekedar menampilkan hasil *output* dari nomor hadist yang hilang serta hasil waktu proses saja, diharapkan untuk kedepannya algoritma ini dapat digunakan dan ada penambahan-penambahan sehingga lebih lengkap dan lebih kompleks.
2. Semoga penelitian ini dapat bermanfaat bagi orang lain, untuk itu mari kita menghargai setiap hasil karya orang lain meskipun kami sadar bahwa masih banyak kekurangan dari penelitian yang telah dilakukan ini, setidaknya dari penelitian yang dilakukan ini sudah ada penambahan dan perbaikan dari sistem pencarian yang pernah dibuat sebelumnya, dan semoga penelitian yang telah dilakukan bisa menjadi tolak ukur untuk penelitian-penelitian selanjutnya.

DAFTAR PUSTAKA

Agustina, Sunita. (2018). *Optimasi Pencarian Kata-Kata Dalam Empat Kitab Hadist*

Arora, S. and Barak, B. (2009). *Computational Complexity: A Modern Approach.* New York: Cambrigde University Press.

Goldreich, Oded. (2008). *Computational Complexity: A Conceptual Perspective.* New York: Cambride University Press.

Hardisman, (2017). *Tuntunan Ahklak Dalam Al-quran Dan Sunnah. Cetakan pertama*. Padang: Andalas University Press.

Higham, N. J.(1998). *Handbook of Writing for the Mathematical Science, (2nd ed).* Piladelphia: SIAM.

Kadir, Abdul dan Heriyanto.. (2005) *Algoritma Pemrograman menggunakan C++*. Yogyakarta:Andi

Liana, Ayu. (2009). *Testing and Implemetation*. Jakarta: Software Engineering.

Munir, Rinaldi. (2011). *Algoritma dan Pemrograman Dalam Bahasa Pascal dan C. Edisi Revisi*. Bandung: Informatika.

Rekrisna, Fembi. (2014). *Paper Struktur Data Dan Algoritma Running-time.* Surakarta: Universitas Sebelas Maret.

Ziyaddurohman, Rifqi. (2018). *Perbandingan Pencarian Hadis Dengan dan Tanpa Index Pada Sistem Pencarian Hadis Riwayat Imam Malik dan Bukhari.*

Setiawan, Wawan. (2013). *Analisis Kinerja Algoritma Pemrograman Dinamik Pada Masalah Multistage Graph.* Malang.

Sidi, Jaka. (2017). *Pengembangan Sistem Pencarian Informasi Pada Hadist Riwayat Bukhari.*

Subandijo, (2011). *Efesiensi Algoritma dan Notasi O-Besar.* Jakarta: Informatika.