

ANALISA PERBANDINGAN ALGORITMA BITAP DAN ALGORITMA ZHU TAKAOKA UNTUK Pencarian String pada Aplikasi AL-QUR'AN BERBASIS MOBILE

Abdi Mulia

Program Studi Teknik Informatika, STMIK Budi Darma, Medan, Indonesia
Jalan Sisingamangaraja No. 338, Medan, Indonesia

Abstrak

Saat ini alat-alat telekomunikasi telah berkembang di Indonesia, salah satu alat telekomunikasi yang sangat pesat perkembangannya saat ini adalah alat telekomunikasi berbasis mobile. Mobile adalah suatu sistem operasi untuk perangkat Android yang berbasis Linux. AL-Qur'an adalah kitab suci yang memuat hadist-hadist yang dijadikan sumber pengambilan hukum fikih (istinbath) oleh para ahli fikih. Kitab ini, juga menjadi rujukan utama khususnya dari fikih dari Mazhab Syafi'i. Karena jumlah Hadist yang tidak sedikit, saat ini Hadist sudah terhimpun bukan satu Hadist saja tetapi sudah ribuan. Maka bagi si pembacapun tidak akan membawa semua buku Hadistnya ketika keluar rumah, dikarenakan Kitab suci berfisik tebal, berbahasa Arab serta belum dilengkapi terjemahan. Sehingga, si pembacapun akan kesusahan untuk mencari masalah mengenai hukum-hukum islam. Proses pencarian merupakan salah satu kegiatan penting dalam pemrosesan data. Proses ini dapat menghabiskan waktu dalam ruang pencarian yang besar sehingga diperlukan suatu teknik pencarian yang efisien. Algoritma Bitap dan Algoritma Zhu takaoka merupakan suatu solusi pencarian yang efisien dapat melakukan perbandingan pattern mulai dari kanan ke kiri. Jika terjadi ketidakcocokan string dari kanan pattern maka ketidakcocokan akan membantu kita untuk menggerakkan pattern tersebut dengan jarak yang lebih jauh. Algoritma Bitap dan Algoritma Zhu takaoka mempunyai keunggulan dalam waktu menemukan pattern yang akan dicari dalam ukuran file yang lebih besar.

Kata Kunci: Mobile, Al-Qur'an, String Matching, Bitap dan Zhu takaoka

I. PENDAHULUAN

Al-Quran merupakan kitab suci umat Islam yang berisikan ilmu pengetahuan tentang hukum-hukum Islam dan kisah umat terdahulu. Sebagai seorang muslim mempelajari Al-Quran adalah suatu kewajiban yang sudah dijalankan secara turun temurun. Al-Quran diturunkan dalam bahasa Arab. Bagi umat muslim di Indonesia mempelajari Al-Quran dilakukan dengan merujuk kepada terjemahan ke Bahasa Indonesia.

Kitab Suci Al-Quran terdiri dari 30 Juz, 114 surah dan 6236 ayat. Dengan dokumen yang begitu banyak, pencarian surat, ayat atau kata yang dilakukan secara manual membutuhkan waktu lama. Seiring perkembangan zaman saat ini dengan menggunakan aplikasi android hal tersebut dapat dilakukan dengan lebih cepat yaitu dengan menerapkan algoritma string matching. Algoritma string matching adalah algoritma yang digunakan untuk pencocokan string sehingga pengguna dapat mencari string atau kata yang diinginkan.

Algoritma *Zhu-Takaoka* yang merupakan modifikasi dari Algoritma *Boyer Moore* mempunyai ciri-ciri yang sama dalam proses pencarian *string*. Ciri-ciri tersebut yaitu terbagi dua *fase* yaitu *fase preprocessing* dan *fase pencarian*. *Zhu-Takaoka* menghitung nilai pergeseran menggunakan *array* dua dimensi dan pencocokan string dari kanan ke kiri. Algoritma Bitap merupakan Algoritma yang bekerja dengan cara melakukan pemetaan menjadi operasi bitwise yang mudah. Setiap bit yang bernilai 0 berarti cocok, dan setiap bit yang bernilai 1 berarti tidak cocok.

Algoritma *Zhu-Takaoka* dan Algoritma Bitap merupakan algoritma yang melakukan pencarian single pattern, Namun Algoritma *Zhu-Takaoka* dan Algoritma

Bitap sama-sama memiliki kelebihan dan kekurangan masing dalam pencocokan string. Untuk mengetahui kelebihan dan kekurangan tersebut demi memilih algoritma yang lebih cepat dalam menentukan penggunaan algoritma yang berfungsi untuk pencarian string penulis melakukan analisa perbandingan algoritma *Zhu-Takaoka* dan algoritma Bitap.

Dengan melakukan analisa perbandingan performansi dari algoritma *Zhu-Takaoka* dan Algoritma Bitap maka akan dapat diketahui cara kerja dan performansi dalam kecepatan dan ketepatan dari kedua algoritma tersebut dan menemukan solusi memilih algoritma yang lebih tepat untuk digunakan pada perangkat lunak pencarian pattern.

II. TEORITIS

A. Algoritma Bitap

Algoritma *Shift-or* merupakan salah satu algoritma *string matching* yang efisien untuk masalah pencocokan *string* secara tepat dan menggunakan metode pencocokan *string* dari kiri ke kanan. Algoritma *Shift-or* ini mempunyai batas ukuran. Ukuran *pattern* yang akan dicocokkan tidak boleh melebihi 32 bit. Batas ukuran tersebut disebabkan karena algoritma ini menggunakan *match register* yang hanya dapat menampung satu *pattern*, sehingga algoritma ini hanya menangani *pattern* dengan panjang maksimal 32 karakter.

Algoritma ini menggunakan sebuah tabel *cancel mask* dan sebuah variabel *match register*. Maka sebelum pencocokan dilakukan, tabel *cancel mask* dan *match register* harus dibangun terlebih dulu. Untuk mengilustrasikan deskripsi dan cara kerja dari algoritma *shift-or* ini, akan digunakan contoh berikut:

1. *pattern: gcagagag.*

2. teks: **gcacgcagagagtatacagtagc.**

Untuk melakukan pencocokan, dilakukan pemrosesan terhadap teks dengan cara:

1. Lakukan operasi logika *OR* terhadap *match register* dan tabel *cancel mask* untuk karakter pertama pada teks (untuk karakter ke n , tabel *cancel mask*-nya adalah seluruh kolom dari baris karakter n tersebut).
2. Simpan hasilnya pada *match register* (setelah dilakukan *shift* kiri pada hasil operasi tadi). Variabel *match register* baru ini menjadi *match register* yang di-*OR*-kan dengan *cancel mask* untuk karakter berikutnya.
3. Lakukan proses di atas pada semua karakter pada teks sampai terdapat 0 pada kolom paling kiri *match register* hasil operasi logika *OR* (yang berarti ditemukan bagian dari teks yang sama dengan *pattern*). [1]

B. Algoritma Zhu Takaoka

Algoritma *Zhu-Takaoka* merupakan salah satu algoritma pencocokan *string* (*String Matching*) yang merupakan pengembangan dari algoritma *Boyer-Moore*. Algoritma ini dipublikasikan oleh Zhu Rui Feng dan Tadao Takaoka pada tahun 1986. Dalam penelitiannya, Zhu dan Takaoka menyebut algoritma pencocokan *string* ini sebagai BM" Algorithm (*Boyer-Moore Algorithm*) karena merupakan modifikasi dari algoritma pencocokan *string* *Boyer-Moore*, yaitu algoritma yang dibuat oleh Boyer R.S. dan Moore J.S. Algoritma *Zhu-Takaoka* dan algoritma *Boyer-Moore* memiliki ciri yang sama dalam proses pencarian *string*, yaitu terdapat tahap *Preprocessing*, *Right-to-left scan*, *Bad character rule* dan *Good-suffix rule*. Sementara itu, perbedaan dari kedua algoritma tersebut terletak pada tahap penentuan *Bad character rule*. Dalam *Boyer-Moore*, *bad character* hanya terdiri dari array satu dimensi, sedangkan dalam *Zhu-Takaoka* array dimodifikasi menjadi dua dimensi.

Adapun tahap pencarian *pattern* menggunakan algoritma *zhu-takaoka* adalah sebagai berikut :

1. *Preprocessing*
Preprocessing dalam algoritma *Zhu-Takaoka* meliputi pencarian nilai pergeseran karakter (*good-suffix shift*) dan pergeseran karakter jika karakter tidak cocok (*bad-character shift*). Nilai *good-suffix shift* ditentukan dalam *good-suffix preprocessing* sedangkan nilai *bad-character shift* ditentukan dalam *bad-character preprocessing*. *Preprocessing* dilakukan sebelum proses inti dari pencarian *pattern* dalam suatu *text*. (Pratama, 2008)
2. *Right-to-Left Scan Rule*
Proses inti pencarian Algoritma *Zhu-Takaoka* yaitu dilakukan dengan teknik *Right-to-left scan rule*. Teknik ini yaitu melakukan perbandingan antara *pattern* yang dicari dengan *target text* secara terbalik yaitu bergerak dari kanan ke kiri. Perbandingan *pattern* dengan *target text* dimulai dengan membandingkan karakter terakhir dari

pattern (karakter paling kanan) dengan *target text* paling kanan. Apabila ada kecocokan maka perbandingan akan dilanjutkan dengan bergerak ke kiri sampai karakter pertama dari *pattern*. Sedangkan apabila terjadi ketidakcocokan maka akan dilakukan pergeseran, besarnya pergeseran yang dilakukan ditentukan oleh dua fungsi pergeseran yaitu *bad-character shift* dan *good-suffix shift*.

3. *Bad-Character Shift Rule*

Aturan *bad-character shift* dibutuhkan untuk menghindari pengulangan perbandingan yang gagal dari suatu karakter dalam *target text* dengan *pattern*. Besarnya pergeseran yang dilakukan dalam aturan *bad-character shift* disimpan dalam bentuk tabel array dua dimensi, tabel ini terdiri dari beberapa kolom yaitu kolom karakter dan kolom *shift* yang menunjukkan besarnya pergeseran yang harus dilakukan.

4. *Good Suffix Shift Rule*

Aturan *good-suffix shift* dibuat untuk menangani kasus dimana terdapat pengulangan karakter pada *pattern*. Contoh dibawah ini akan menjelaskan bagaimana aturan *bad-character shift* gagal dalam menangani adanya pengulangan bagian dalam *pattern*. [2]

C. AL-Qur'an

Menurut Dr. Subhi as-Salih, Al Qur'an adalah kalam Allah swt merupakan mukjizat yang diturunkan kepada Nabi Muhammad saw ditulis dalam mushaf dan diriwayatkan dengan mutawatir serta membacanya adalah ibadah.

III. ANALISA DAN PEMBAHASAN

Sebagai contoh, akan dilakukan proses pencarian *pattern* menggunakan algoritma *Bitap* dengan perbandingan dari kanan ke kiri : *Text* "BISMILLAH" dan *Pattern* "BIS".

Langkah-langkah Algoritma yaitu:

1. Algoritma dimulai dengan melakukan perhitungan *bitmask* yang mengandung bit untuk setiap elemen dalam pola.
2. kemudian pencarian akan dilakukan dengan cara operasi *bitwise*, dimana pencarian tersebut sangat cepat dilakukan.
3. Bekerja dengan cara melakukan pemetaan menjadi operasi *bitwise* yang mudah dan Setiap bit yang bernilai 0 berarti cocok, dan setiap bit yang bernilai 1 berarti tidak cocok.

Perancangan aplikasi pada penelitian ini dilakukan dengan menggunakan *software IDE Eclipse*. Editor kode program dan android SDK (*software development kit*) yang digunakan untuk mengembangkan aplikasi ini menggunakan bahasa pemrograman Java serta *Android Development Tools (ADT)*. Setelah melakukan pengkodean hasil dari perancangan aplikasi ini dapat dioperasikan pada

emulator di laptop dan *smartphone* yang didukung oleh sistem operasi *android*.

Pergeseran Ke-1

Index	1	2	3	4	5	6	7	8	9
Text	B	I	S	M	I	L	L	A	H
Pattern						B	I	S	

Pada pattern "s" jika tidak cocok dengan index "h" maka geser pattern satu langkah ke kiri menuju index berikutnya.

Pergeseran Ke-2

Index	1	2	3	4	5	6	7	8	9
Text	B	I	S	M	I	L	L	A	H
Pattern					B	I	S		

Pada pattern "s" jika tidak cocok dengan index "a" maka geser pattern satu langkah ke kiri menuju index berikutnya.

Pergeseran Ke-3

Index	1	2	3	4	5	6	7	8	9
Text	B	I	S	M	I	L	L	A	H
Pattern				B	I	S			

Pada pattern "s" jika tidak cocok dengan index "i" maka geser pattern satu langkah ke kiri menuju index berikutnya.

Pergeseran Ke-4

Index	1	2	3	4	5	6	7	8	9
Text	B	I	S	M	I	L	L	A	H
Pattern			B	I	S				

Pada pattern "s" jika tidak cocok dengan index "i" maka geser pattern satu langkah ke kiri menuju index berikutnya.

Pergeseran Ke-5

Index	1	2	3	4	5	6	7	8	9
Text	B	I	S	M	I	L	L	A	H
Pattern		B	I	S					

Pada pattern "s" jika tidak cocok dengan index "m" maka geser pattern satu langkah ke kiri menuju index berikutnya.

Pergeseran Ke-6

Index	1	2	3	4	5	6	7	8	9
Text	B	I	S	M	I	L	L	A	H
Pattern	B	I	S						

Pada pattern "s" cocok dengan index "s", "i" cocok dengan "i", "b" cocok dengan "b" maka pattern ditemukan dan pencarian berhenti di indeks ke 6.

Adapun contoh kasus pencarian pattern menggunakan algoritma *zhu-takaoka* adalah sebagai berikut :

- Langkah pertama yang dilakukan Algoritma *zhu takaoka* tahapan *preprocessing* yang menciptakan 2 buah table pergeseran *ztBc* (*Zhu takaoka Bad Character*) dan *bmGs* (*Boyer Moore Good Suffix*).

- Kedua table ini di ciptakan tergantung pattern yang akan dicari oleh karna itu jika *pattern* yang dimasukan berubah maka tabel juga akan berubah.
- Hasil *preprocessing* untuk *patern* "Bis" terlihat pada tabel dibawah

Teks = bismillah

Pattern = bis

Tabel 1. Zhu-Takaoka Bad Character

ZtBc	B	I	S	*
B	2	3	1	3
I	2	3	3	3
S	2	3	3	3
*	2	3	3	3

Tabel *ztBc* berbentuk array dua dimensi yang baris dan kolom diisi sesuai dengan karakter yang ada pada pattern, tanda *(start) digunakan untuk mewakili seluruh karakter yang tidak ada pada pattern. Tabel *ztBc* merupakan hasil modifikasi dari algoritma Boyer Moore yang memiliki table bad character hanya terdiri dari array satu dimensi. Dapat dilihat pada tabel di bawah ini:

Tabel 2. Boyer-Moore Good Suffixes

I	0	1	2
$x[i]$	B	I	S
$suff[i]$	0	0	3
$bmGs[i]$	3	3	1

Setelah menemukan tabel *ztBc* tabel *bmGs* langkah selanjutnya yaitu melakukan pencocokan pattern dengan teks.

Adapun langkah-langkah pencarian pattern dengan algoritma *Zhu-Takaoka* dapat dilihat pada tabel dibawah ini:

Langkah 1

Pencarian pada teks langkah ke - 1

Window	B	I	S							
Text	I	S	M	I		B	I	S	M	I
Pattern	B	I	S							
I	0	1	2							

$ztBc[B][I] = 3$

$bmGs[i] = bmGs[2] = 1$

Pergeseran dilakukan sebanyak 1

Langkah 2

Pencarian pada teks langkah ke – 2

[illegible]
$$\mathbf{ztBc}[S][E] = 3$$

Karakter cocok semua

Pergeseran dilakukan sebanyak **bmGs[0] = 3**

Langkah 3

Pencarian pada teks langkah ke - 3

<i>Window</i>				B	I							
<i>Text</i>	B	I	S	M	B	I	S	M	I	L	L	A H
<i>Pattern</i>					B	I	S					
<i>I</i>					0	1	2					

$$\text{ztBc}[S][E] = 3$$
$$\mathbf{bmGs}[i] = \mathbf{bmGs}[0] = 3$$

Pergeseran dilakukan sebanyak 3

Langkah 4

Pencarian pada teks langkah ke - 4

Window															
Text	B	I	S	M		B	I	S	M	I	L	L	A	H	
Pattern						B	S	S							
I						0	1	2							

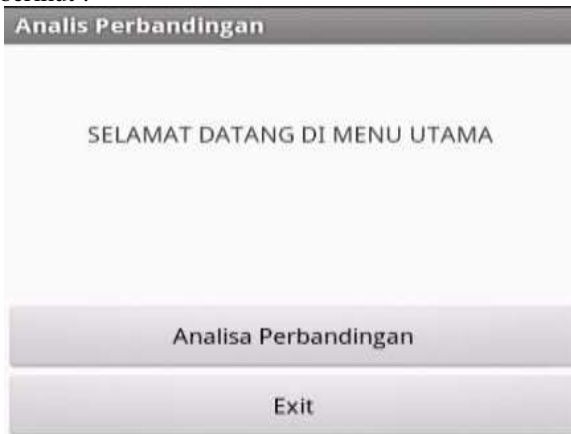
Karakter cocok semua.

Pergeseran dilakukan sebanyak **bmGs[0] = 2**

Karena panjang teks sudah habis maka proses pencocokan dihentikan. Berdasarkan contoh diatas penulis menarik kesimpulan bahwa dengan menggunakan text BIS BISMILAH dan pattern BIS menggunakan Algoritma Zhu-Takaoka mengalami dua kecocokan pola.

IV. IMPLEMENTASI

Untuk menguji aplikasi Mobile Alqur'an, maka perlu mempersiapkan sistem yang terdiri dari *hardware* (perangkat keras), dan *software* (perangkat lunak). Saat pertama kali dijalankan akan muncul tampilan sebagai berikut :



Gambar 1. Tampilan Menu Utama

Tampilan gambar 1 berisi Pencarian perbandingan yang berhubungan dengan ketersediaan,

dimana pengguna dapat melakukan pencarian Algoritma dengan cara memasukan kata kedalam textbox yang tersedia di aplikasi.



Analisa perbandingan zhu takaoka dan bitap

Input Pattern

Search


Output Pattern

Jumlah Langkah Pencarian zhu takaoka

Jumlah Langkah Pencarian Bitap

Gambar 2. Tampilan pencarian Bitap dan Zhu Takaoka

Tampilan gambar 2 berisi Pencarian perbandingan yang berhubungan dengan ketersediaan, dimana pengguna dapat melakukan pencarian Algoritma dengan cara memasukan kata kedalam textbox yang tersedia di aplikasi



The screenshot shows the Bitap application interface. At the top, there is a title bar with the text "Analisa perbandingan zhu takaoka dan bitap". Below the title bar, there is a section labeled "Input Pattern" with a text input field containing "Bismillah" and a "Search" button. Below the input field, there is a section labeled "Output Pattern" with a text input field containing "Dengan menyebut nama Allah". Below the output field, there are two sections: "Jumlah Langkah Pencarian zhu takaoka" with a value of "109000" and "Jumlah Langkah Pencarian Bitap" with a value of "437000".

Analisa perbandingan zhu takaoka dan bitap

Input Pattern

Bismillah

Search

Output Pattern

Dengan menyebut nama Allah

Jumlah Langkah Pencarian zhu takaoka

109000

Jumlah Langkah Pencarian Bitap

437000

Gambar 3. Hasil pencarian Algoritma Bitap dan Algoritma Zhu Takaoka

Tampilan gambar 3 berisi tentang informasi tentang aplikasi dimana pengguna dapat melihat *profil* dan menampilkan hasil pencarian.

V. KESIMPULAN

Berdasarkan pembahasan-pembahasan sebelumnya serta dari pengujian-pengujian yang telah dilakukan, maka dapat diambil beberapa kesimpulan yaitu :

1. Dengan ini disimpulkan proses pencarian kata atau *pattern* Algoritma Bitap dan Algoritma Zhu Takaoka
2. Cara kerja Algoritma Bitap dan Algoritma Zhu Takaoka dalam pencarian Al-Qur'an adalah mencari dengan cara membandingkan sebuah huruf dengan huruf yang ada di *pattern* yang dicari dan *massager pattern* tersebut hingga posisinya sama dengan teks yang dicari dan membandingkan dengan kata tersebut.
3. Hasil akhir dari penelitian ini adalah sebuah aplikasi AL-Qur'an berbasis Mobile yang dikembangkan dengan Eclipse Juno.

REFERENCES

- [1] Fikri Rijul, dkk, pemograman java, Andi, yogyakarta, 2005.
- [2] Nasruddin Safaat H, Pemograman Aplikasi Mobile Smarphone dan Table PC Berbasis Android, Informatika Bandung,Bandung,2015.
- [3] A,S Rosa dan M. Shalahuddin, Rekayasa perangkat Lunak , Modula, yogyakarta, 2011.
- [4] Alfa Satyaputra dan Eva Maulina Aritonang, java for Beginners With Eclipse 4.2 junio, PT.Elex Media Komputindo, jakarta,2012.
- [5] J.I. Sinaga, Mesran, E. Buulolo, APLIKASI MOBILE PENCARIAN KATA PADA ARTI AYAT AL-QUR'AN BERBASIS ANDROID MENGGUNAKAN ALGORITMA STRING MATCHING, INFOTEK. 2 (2016) 68–72.