

# Perbandingan Algoritma Boyer-Moore dan Brute Force pada Pencarian Kamus Besar Bahasa Indonesia Berbasis Android

*by* Mudafiq Riyan Pratama

---

**Submission date:** 22-Feb-2021 11:37AM (UTC+0700)

**Submission ID:** 1514882319

**File name:** 13-Article\_Text-18-2-10-20210211.pdf (875.19K)

**Word count:** 3153

**Character count:** 16817

## Perbandingan Algoritma Boyer-Moore dan *Brute Force* pada Pencarian Kamus Besar Bahasa Indonesia Berbasis Android

Candra Irawan<sup>1</sup>, Mudafiq Riyan Pratama<sup>2</sup>

<sup>1</sup>Teknik Informatika, Universitas Muhammadiyah Jember, [candwan66@gmail.com](mailto:candwan66@gmail.com)

<sup>2</sup>Politeknik Negeri Jember, [mudafiq.riyan@polije.ac.id](mailto:mudafiq.riyan@polije.ac.id)

### Keywords:

String Matching,  
Boyer-Moore,  
Brute Force,  
Searching,  
KBBI,

### ABSTRACT

String matching is an algorithm for matching a text to another text or also known as a text search. There are several algorithms that can be used for string matching, including the Boyer-Moore algorithm and the Brute Force algorithm. The Boyer-Moore algorithm is a string matching algorithm published by Robert S. Boyer and J. Strother Moore in 1977. This algorithm is considered the most efficient algorithm in general applications. The Boyer-Moore algorithm starts matching characters from the pattern from the right. While the Brute Force algorithm is an algorithm that matches a pattern with all text between 0 and n-m to find the existence of a pattern in the text. These two algorithms have different patterns in the search process. In this article, a comparative analysis of the performance of the Boyer-Moore and Brute Force algorithms is carried out in a case study of the search for the Big Indonesian Dictionary (KBBI) based on Android. The search process is carried out by searching based on words and word descriptions. The results of this study indicate that the criteria for running time, the Brute Force algorithm is faster than the Boyer-Moore algorithm with the total running time of the Brute Force algorithm is 168.3 ms in words, 6994.16 ms in word descriptions, while the Boyer-Moore algorithm for running time reached 304.7 ms on the word, 8654.77 ms on the word description. In the testing criteria based on related keywords, the two algorithms can display the same list of related keywords.

### Kata Kunci

String Matching,  
Boyer-Moore,  
Brute Force,  
Pencarian,  
KBBI

### ABSTRAK

String matching merupakan algoritma untuk pencocokan suatu teks terhadap teks lain atau juga disebut sebagai pencarian teks. Ada beberapa algoritma yang dapat digunakan untuk string matching, diantaranya algoritma Boyer-Moore dan Brute Force. Algoritma Boyer-Moore adalah salah satu algoritma pencocokan string yang dipublikasikan oleh Robert S. Boyer dan J. Strother Moore pada tahun 1977. Algoritma ini dianggap sebagai algoritma yang paling efisien pada aplikasi umum. Algoritma Boyer-Moore memulai pencocokan karakter dari pattern sebelah kanan. Sedangkan algoritma Brute Force adalah algoritma yang mencocokkan pattern dengan semua teks antara 0 dan n-m untuk menemukan keberadaan pattern dalam teks. Kedua algoritma ini memiliki pola yang berbeda dalam proses pencariannya. Pada artikel ini dilakukan analisa perbandingan kinerja algoritma Boyer-Moore dan Brute Force pada studi kasus pencarian Kamus Besar Bahasa Indonesia (KBBI) berbasis Android. Proses pencarian dilakukan dengan melakukan pencarian berdasarkan kata dan deskripsi kata. Hasil penelitian ini menunjukkan bahwa pada kriteria running time, algoritma Brute Force lebih cepat dibandingkan algoritma Boyer-Moore dengan total running time algoritma Brute Force adalah 168,3 ms pada kata, 6994,16 ms pada deskripsi kata, sedangkan pada algoritma Boyer-Moore untuk running time mencapai 304,7 ms pada kata, 8654,77 ms pada deskripsi kata. Pada kriteria pengujian berdasarkan keyword related, kedua algoritma dapat menampilkan daftar keyword related yang sama.

### Korespondensi Penulis:

Mudafiq Riyan Pratama  
Politeknik Negeri Jember  
Jl. Mastrip 164 Jember  
Email: [mudafiq.riyan@polije.ac.id](mailto:mudafiq.riyan@polije.ac.id)

1. PENDAHULUAN

String matching merupakan suatu algoritma yang digunakan untuk pencocokan suatu teks terhadap teks lain atau juga disebut sebagai pencarian teks. Ada beberapa algoritma yang dapat digunakan untuk string matching, diantaranya adalah algoritma Boyer-Moore [1] dan Brute Force. Algoritma Boyer-Moore adalah salah satu algoritma pencarian string yang dipublikasikan oleh Robert S. Boyer dan J. Strother Moore pada tahun 1977. Algoritma Boyer-Moore dianggap sebagai salah satu algoritma yang paling efisien untuk aplikasi pencocokan pola umum. Ia mampu mengenali dan melewati area tertentu dalam teks yang tidak ditemukan kecocokan [2]. Ide dibalik algoritma ini adalah bahwa dengan memulai pencocokkan karakter dari kanan, dan bukan dari kiri, maka akan lebih banyak informasi yang didapat [3].

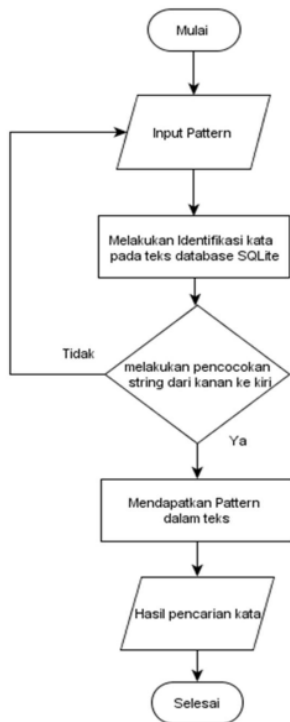
Sedangkan Algoritma Brute Force adalah algoritma pattern matching yang paling sederhana, dan pencarian Brute Force bersifat left-to-right, yakni dari kiri ke kanan, sehingga pattern akan dicocokkan huruf per huruf dalam sebuah teks, dan ketika sebuah huruf dalam pattern yang ingin dicocokkan tidak sesuai dengan salah satu huruf dalam teks, maka pencarian akan diulang dan dimulai pada huruf teks selanjutnya [4]. Kedua algoritma memiliki cara pencarian yang berbeda, sehingga tentunya akan menghasilkan kecepatan dan hasil yang berbeda.

Dengan adanya perbedaan cara kerja kedua algoritma tersebut, maka perlu adanya analisa yang membandingkan kinerja kedua algoritma tersebut dalam pencarian kata pada KBBI (Kamus Besar Bahasa Indonesia). KBBI ini menjadi pilihan karena berisi lexicon atau kamus kata yang jumlahnya besar, sehingga semakin banyak data yang dicari, akan terlihat kinerja dari masing-masing algoritma tersebut.

2. METODE PENELITIAN

2.1 Flowchart Sistem

Algoritma yang dipilih pada penelitian ini ada dua, yaitu algoritma Boyer-Moore dan Brute Force. Flowchart dari masing-masing algoritma tersebut digambarkan pada Gambar 1 dan Gambar 2.

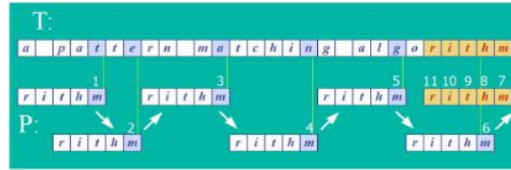


Gambar 1. Flowchart Boyer-Moore



Gambar 2. Flowchart Brute Force

Dari Gambar 1 di atas, dijelaskan alur proses pencarian teks menggunakan algoritma Boyer-Moore, yaitu pertama user memasukkan pattern berupa karakter yang ingin dicari di database KBBI. Kemudian melakukan identifikasi kata yang ada di database dengan melakukan pencocokan string dari kanan ke kiri sesuai dengan inputan pattern yang diinginkan oleh user. Proses tersebut melakukan looping atau perulangan sampai semua data di database ditemukan kecocokan dengan pattern yang dicari. Dan sistem akan menampilkan output dari hasil pencarian. Berikut ini contoh alur dari algoritma Boyer-Moore [5]



Gambar 3. Contoh alur dari algoritma Boyer-Moore

Secara sistematis, langkah-langkah yang dilakukan algoritma Boyer-Moore pada saat mencocokkan string adalah [6]:

1. Algoritma Boyer-Moore mulai mencocokkan pattern pada awal teks.
2. Dari kanan ke kiri, algoritma ini akan mencocokkan karakter per karakter *pattern* dengan karakter di teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi:
  - a. Karakter di pattern dan di teks yang dibandingkan tidak cocok (*mismatch*).
  - b. Semua karakter di pattern cocok. Kemudian algoritma akan memberitahukan penemuan di posisi ini.
3. Algoritma kemudian menggeser *pattern* dengan memaksimalkan nilai penggeseran *good-suffix* dan penggeseran *bad-character*, lalu mengulangi langkah 2 sampai *pattern* berada di ujung teks.

Sedangkan pada Gambar 2 merupakan penjelasan alur dari algoritma *Brute Force*, yaitu user memasukkan *pattern* yang ingin dicari, kemudian sistem melakukan pembacaan *pattern* pada database KBBI yang dilakukan pencocokan *string* dari kiri ke kanan. Sistem akan melakukan *looping* pencocokan *string* dan menampilkan data yang sesuai dengan *pattern* yang dicari. Secara sistematis, langkah-langkah yang dilakukan algoritme *brute force* pada saat mencocokkan string adalah [4]:

1. Algoritme *Brute Force* mulai mencocokkan pattern pada awal teks.
2. Dari kiri ke kanan, algoritma ini akan mencocokkan karakter per karakter *pattern* dengan karakter di teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi:
  - a. Karakter di pattern dan di teks yang dibandingkan tidak cocok (*mismatch*).
  - b. Semua karakter di *pattern* cocok. Kemudian algoritma akan memberitahukan penemuan di posisi ini.
3. Algoritma kemudian terus menggeser *pattern* sebesar satu ke kanan, dan mengulangi langkah ke-2 sampai *pattern* berada di ujung teks.

## 2.2 Skenario Pengujian

Pengujian dilakukan untuk membandingkan kinerja dari algoritma Boyer-Moore dan *Brute Force* yang dilakukan dengan dua skenario yaitu pencarian pada kata dan pencarian pada deskripsi kata yang masing-masing dilakukan pengujian berdasarkan *running time* (kecepatan proses pencarian) dan *keyword related* (kata yang muncul dari hasil pencarian).

## 3. HASIL DAN ANALISIS

Pencarian dilakukan berdasarkan *string pattern* yang diketikkan pada kolom pencarian yang kemudian string tersebut dicocokkan pada database KBBI yang berjumlah 17.000 kata. Setiap proses akan dilakukan pencarian pada kata dan deskripsi kata yang telah tersedia pada database KBBI, kemudian dihitung jumlah *running time* dan *keyword related* nya. Hasil pengujian tersebut dipaparkan pada subbab berikut ini:

### 3.1 Hasil Pengujian Pencarian Kata dengan Algoritma Boyer-Moore

Pada pengujian ini dilakukan pencarian dengan 20 kata kunci menggunakan algoritma Boyer-Moore. Dan setiap kata kunci yang dicari akan melakukan pencarian pada data kata dan deskripsi kata yang tersedia di KBBI. Hasil dari pengujian ini dipaparkan pada tabel di bawah ini.

Tabel 1. Hasil pengujian dengan algoritma Boyer-Moore

No.	Pattern	Running Time (ms)		Keyword related	
		Kata	Deskripsi	Kata	Deskripsi
1	an	234,08	2658,62	2890	16459
2	me	33,93	2219,75	857	12091
3	di	22,38	1696,5	618	10162
4	dia	2,83	600,93	64	1909
5	aku	3,35	530,46	79	1705
6	aki	2,43	618,3	56	1875
7	auto	0,57	0,53	14	7
8	beli	0,99	119,73	27	367
9	buku	0,43	64,91	4	252
10	nasi	1,58	101,52	48	512

11	verbal	0,25	2,09	7	11
12	pasca	0,42	0,32	11	5
13	pramu	0,36	2,68	10	23
14	humor	0,18	0,46	4	6
15	profesi	0,24	5,23	4	36
16	prospek	0,18	0,22	4	2
17	spesial	0,19	0,28	4	2
18	seismogra	0,12	0,13	2	1
19	perempuan	0,08	31,93	1	245
20	harmonika	0,11	0,18	1	2
<b>Total</b>		<b>304,7</b>	<b>8654,77</b>	<b>4705</b>	<b>45672</b>
<b>Rata-rata</b>		<b>15,235</b>	<b>432,7</b>		

Dari tabel di atas, pada 20 kali percobaan pencarian *pattern*, diperoleh total *running time* pada pencarian kata adalah 304.7 ms, sedangkan total *running time* pada pencarian deskripsi adalah 8654.77 ms. Hasil *keyword related* yang ditemukan pada kata berjumlah total 4705 dan pada deskripsi berjumlah total 45672 kata.

**3.2 Hasil Pengujian Pencarian Kata dengan Algoritma Brute Force**

Pengujian dilakukan dengan *pattern* yang sama seperti pengujian pada subbab 3.1 diatas. Dan hasil yang didapat dengan menggunakan algoritma *Brute Force* dapat dilihat pada tabel di bawah ini

Tabel 2. Hasil pengujian dengan algoritma *Brute Force*

No.	Pattern	Running Time (ms)		Keyword related	
		Kata	Deskripsi	Kata	Deskripsi
1	an	131,3	2409,29	2890	16459
2	me	16,84	1045,81	857	12091
3	di	12,24	1607,88	618	10162
4	dia	1,74	550,62	64	1909
5	aku	1,76	458,6	79	1705
6	aki	1,31	514,98	56	1875
7	auto	0,27	0,56	14	7
8	beli	0,57	155,01	27	367
9	buku	0,21	82,44	4	252
10	nasi	0,78	100,24	48	512
11	verbal	0,15	2,21	7	11
12	pasca	0,26	0,29	11	5
13	pramu	0,2	2,65	10	23
14	humor	0,12	0,43	4	6
15	profesi	0,11	9,57	4	36
16	prospek	0,13	0,35	4	2
17	spesial	0,12	0,21	4	2
18	seismogra	0,1	0,15	2	1
19	perempuan	0,08	51,69	1	245
20	harmonika	0,06	0,18	1	2
<b>Total</b>		<b>168,3</b>	<b>6994,16</b>	<b>4705</b>	<b>45672</b>
<b>Rata-rata</b>		<b>8,415</b>	<b>349,7</b>		

Dari tabel di atas, pada 20 kali percobaan pencarian *pattern*, diperoleh total *running time* pada pencarian kata adalah 168.3 ms, sedangkan total *running time* pada pencarian deskripsi adalah 6994.16 ms. Hasil *keyword related* yang ditemukan pada kata berjumlah total 4705 dan pada deskripsi berjumlah total 45672 kata.

**3.3 Hasil Pengujian Pencarian Kata Palindrom dengan Algoritma Boyer-Moore dan Brute Force**

Kata palindrom merupakan kata yang dapat dibaca dengan sama, baik dari kanan maupun dari kiri. Pengujian ini perlu dilakukan untuk membedakan antara algoritma Boyer-Moore dan *Brute Force* karena kedua algoritma ini memiliki perbedaan arah penelusurannya, yaitu Boyer-Moore melakukan penelusuran dari kanan ke kiri, sedangkan *Brute Force* melakukan penelusuran dari kiri ke kanan. Pengujian dilakukan pada 5 kali percobaan kata. Hasil dari pengujian ini dapat dilihat pada tabel di bawah ini.

Tabel 3. Hasil pengujian pencarian kata palindrom dengan algoritma Boyer-Moore dan *Brute Force*

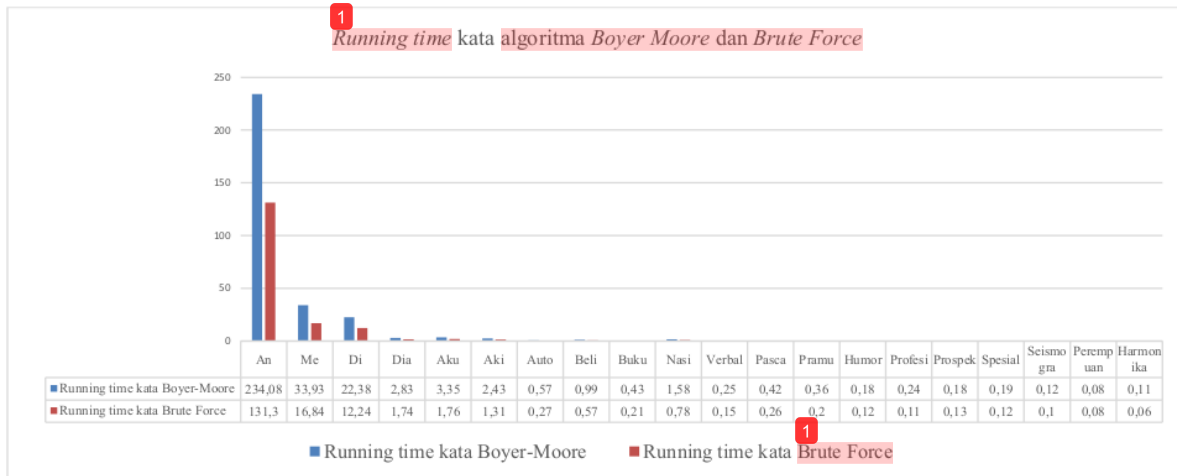
		Running Time (ms)
--	--	-------------------

No	Pattern	Boyer-Moore		Brute Force		Keyword Related Boyer-Moore dan Brute Force	
		Kata	Deskripsi	Kata	Deskripsi	Kata	Deskripsi
1	ada	5,34	555,21	2,63	462,69	157	3499
2	apa	3,56	569,19	1,86	496,97	100	4228
3	taat	0,17	6,49	0,8	5,48	2	26
4	makam	0,47	117,75	0,33	100,4	5	833
5	radar	0,13	2,96	0,8	2,93	2	13
<b>Total</b>		<b>9,67</b>	<b>1251,6</b>	<b>6,42</b>	<b>1068,47</b>	<b>266</b>	<b>8599</b>
<b>Rata-rata</b>		<b>1,93</b>	<b>250,32</b>	<b>1,28</b>	<b>213,69</b>		

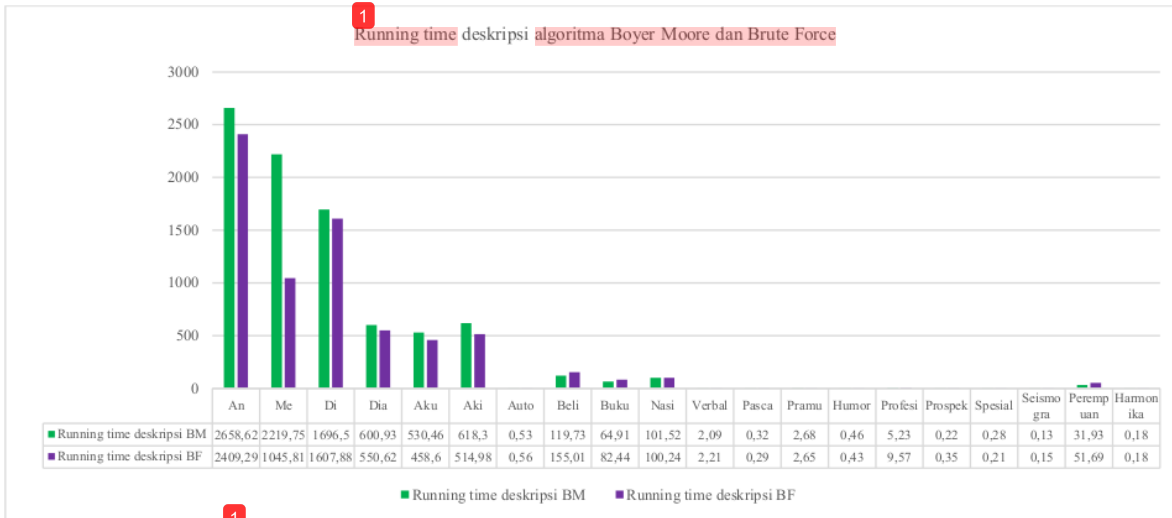
Dapat dilihat pada tabel di atas, hasil pencarian kata palindrom menunjukkan bahwa algoritma *Brute Force* tetap lebih unggul dibandingkan algoritma Boyer-Moore dari sisi *running time*, yaitu *running time* kata pada *Brute Force* totalnya 6.42 ms, sedangkan Boyer-Moore totalnya 9.67 ms. Pun demikian pada total *running time* pada hasil pencarian deskripsi juga lebih cepat algoritma *Brute Force* dibandingkan Boyer-Moore. Sedangkan berdasarkan *keyword related* didapatkan hasil yang sama antara kedua algoritma tersebut.

### 3.4 Analisa Perbandingan Algoritma Boyer-Moore dan Brute Force

Setelah melakukan pengujian pada masing-masing algoritma, perbandingan antara kedua algoritma tersebut dapat dilihat pada gambar di bawah ini.

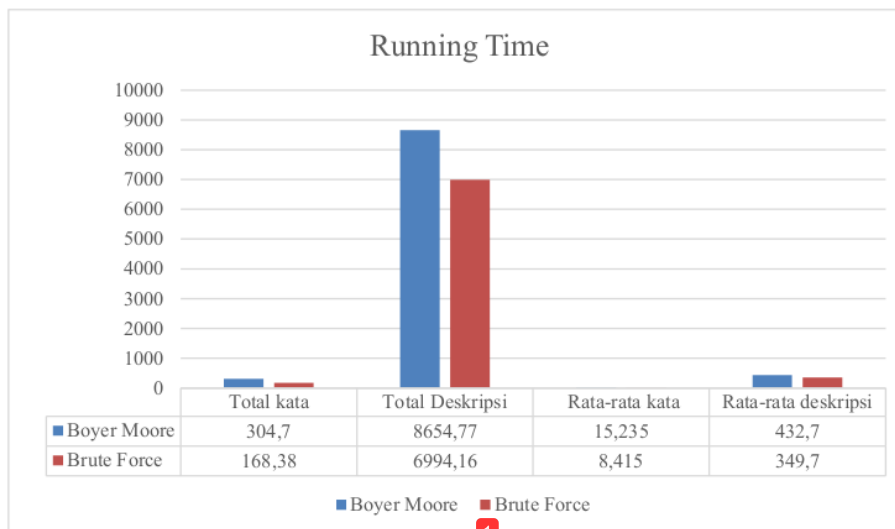


Gambar 4. Grafik perbandingan *running time* pencarian kata pada algoritma Boyer-Moore dan *Brute Force*



Gambar 5. Grafik perbandingan *running time* pencarian deskripsi pada algoritma Boyer-Moore dan *Brute Force*

Berdasarkan grafik di atas, dapat dilihat bahwa *running time* pada pencarian kata maupun deskripsi, secara garis besar, algoritma *Brute Force* lebih cepat dibandingkan Boyer-Moore, meskipun pada beberapa kata kunci berikut: “auto”, “beli”, “buku”, “verbal”, “profesi”, “prospek”, “seismogra”, “perempuan” pada pencarian deskripsi, algoritma *Brute Force* memiliki *running time* yang lebih lama dibandingkan Boyer-Moore. Untuk lebih memastikan mana yang lebih cepat diantara kedua algoritma tersebut, maka dapat dilihat grafik total *running time* pada Gambar 5 di bawah ini.



Gambar 6. Total dan rata-rata *running time* algoritma Boyer-Moore dan *Brute Force*

Antara algoritma Boyer-Moore dan *Brute Force* terdapat perbedaan yang cukup signifikan dari sisi *running time*. Berdasarkan *running time*, bisa disimpulkan bahwa algoritma *Brute Force* dengan total *running time* 168.3 ms (pada pencarian kata) dan 6994.16 ms (pada pencarian deskripsi) lebih cepat dibandingkan algoritma Boyer-Moore yang total *running time* nya 304.17 ms (pada pencarian kata) dan 8654.77 ms (pada pencarian deskripsi). Sedangkan dari sisi *word related*, kedua algoritma memiliki hasil yang sama.

Dari grafik diatas dapat disimpulkan bahwa algoritma *Brute Force* memiliki nilai total dan rata-rata *running time* yang lebih rendah dibandingkan dengan algoritma Boyer-Moore, artinya bahwa algoritma *Brute Force* lebih cepat dibandingkan dengan algoritma Boyer-Moore. Algoritma *Brute Force* lebih cepat karena alurnya pencariannya

dari kiri ke kanan dan akan berhenti ketika telah ditemukan kesesuaian antara *pattern* dengan teks. Dikuatkan lagi ketika dilakukan pengujian pada *pattern* palindrom, pencarian dengan *Brute Force* tetap lebih cepat dibandingkan Boyer-Moore.

Sedangkan dari sisi *keyword related*, kedua algoritma dapat menemukan semua kata yang dicari dengan jumlah yang sama, artinya kedua algoritma tersebut memiliki kualitas yang sama mengenai hasil pencarian. Sehingga tidak ada perbedaan kinerja kedua algoritma tersebut berdasarkan *keyword related*.

#### 4. KESIMPULAN

Dari hasil penelitian ini, dapat disimpulkan beberapa poin berikut:

1. Algoritma *Brute Force* melakukan pencocokan *pattern* dengan teks dari kiri-ke-kanan, sedangkan algoritma Boyer-Moore melakukan pencocokan dari kanan-ke-kiri.
2. Algoritma *Brute Force* memiliki nilai total dan rata-rata *running time* yang lebih rendah dibandingkan dengan algoritma Boyer-Moore, artinya bahwa algoritma *Brute Force* lebih cepat dibandingkan dengan algoritma Boyer-Moore. Algoritma *Brute Force* lebih cepat karena alurnya pencariannya dari kiri ke kanan dan akan berhenti ketika telah ditemukan kesesuaian antara *pattern* dengan teks. Dikuatkan lagi ketika dilakukan pengujian pada *pattern* palindrom, pencarian dengan *Brute Force* tetap lebih cepat dibandingkan Boyer-Moore.
3. Pada algoritma Boyer Moore maupun *Brute Force*, semakin sedikit karakter *pattern* yang dicari maka pencariannya semakin lama karena kata yang dicari semakin banyak dan kata yang ditemukan juga semakin banyak, dan semakin banyak karakter *pattern* yang dicari maka pencariannya semakin cepat karena yang dicari semakin sedikit dan kata yang ditemukan juga semakin sedikit.
4. Dari sisi *keyword related*, kedua algoritma dapat menemukan semua kata yang dicari dengan jumlah yang sama, artinya kedua algoritma tersebut memiliki kualitas yang sama mengenai hasil pencarian.

#### 10 REFERENSI

- [1] R. S. Boyer and J. S. Moore, "A Fast String Searching Algorithm," *Commun. ACM*, vol. 20, no. 10, pp. 762–772, 1977.
- [2] T. Bell, M. Powell, A. Mukherjee, and D. Adjero, "Searching BWT compressed text with the Boyer-Moore algorithm and binary search," *Data Compression Conf. Proc.*, vol. 2002-Janua, pp. 112–121, 2002.
- [3] Tomy Satria Alasi, "Algoritma Boyer Moore Untuk Penyaringan Pesan Teks Menggunakan Perbandingan Kata Yang Sama," *Semin. Nas. Sains Teknol. Inf.*, pp. 488–495, 2018.
- [4] A. Abimanyu, "Penerapan Algoritma String Matching pada Proses Penyaringan Teks," 2019.
- [5] M. Y. Soleh, "Implementasi Algoritma KMP dan Boyer-Moore dalam Aplikasi Search Engine Sederhana," *Makal. IF3051 Strateg. Algoritm.*, 2010.
- [6] S. Kristanto, A. Rachmat, and R. G. Santosa, "Implementasi Algoritma Boyer-Moore Pada Permainan Word Search Puzzle," no. September 2014, pp. 1–8, 2013.



# Perbandingan Algoritma Boyer-Moore dan Brute Force pada Pencarian Kamus Besar Bahasa Indonesia Berbasis Android

## ORIGINALITY REPORT

19%

SIMILARITY INDEX

20%

INTERNET SOURCES

10%

PUBLICATIONS

9%

STUDENT PAPERS

## PRIMARY SOURCES

1	<a href="http://text-id.123dok.com">text-id.123dok.com</a> Internet Source	5%
2	<a href="http://www.scribd.com">www.scribd.com</a> Internet Source	3%
3	<a href="http://bukanpowerrangers.wordpress.com">bukanpowerrangers.wordpress.com</a> Internet Source	3%
4	<a href="http://123dok.com">123dok.com</a> Internet Source	2%
5	<a href="http://ejournal.unmus.ac.id">ejournal.unmus.ac.id</a> Internet Source	1%
6	<a href="http://ukdw.ac.id">ukdw.ac.id</a> Internet Source	1%
7	<a href="http://repository.unmuhjember.ac.id">repository.unmuhjember.ac.id</a> Internet Source	1%
8	<a href="http://id.123dok.com">id.123dok.com</a> Internet Source	1%
9	I Made Agus Wirawan, Ida Bagus Made Ludy	

---

Paryatna. "Implementation of the String Matching Method on Anggah-Ungguhing Balinese Language Dictionary", International Journal of Interactive Mobile Technologies (iJIM), 2020

Publication

1%

---

10

media.neliti.com

Internet Source

1%

---

Exclude quotes On

Exclude matches < 1%

Exclude bibliography On