

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>, Victor Wahanggara<sup>3</sup>

[Candwan66@gmail.com](mailto:Candwan66@gmail.com)<sup>1</sup>; [mudafiq.riyan@unmuhjember.ac.id](mailto:mudafiq.riyan@unmuhjember.ac.id)<sup>2</sup>

[VictorWahanggara@unmuhjember.ac.id](mailto:VictorWahanggara@unmuhjember.ac.id)<sup>3</sup>

Teknik Informatika Universitas Muhammadiyah Jember

Jln. Karimata No. 49, Telp (0331)336728, Jember

**ABSTRAK**

*String matching* adalah suatu algoritma yang digunakan untuk memecahkan masalah pencocokan suatu *teks* terhadap suatu *teks* lain. Ada beberapa algoritma yang dapat digunakan dalam pencarian *string matching* contohnya ialah algoritma *Boyer Moore* dan *Brute Force*. Algoritma *Boyer-Moore* adalah salah satu algoritma pencarian *string*, dipublikasikan oleh Robert S. Boyer, dan J. Strother Moore pada tahun 1977. Algoritma ini dianggap sebagai algoritma yang paling efisien pada aplikasi umum. Tidak seperti algoritma pencarian *string* yang ditemukan sebelumnya, algoritma *Boyer-Moore* mulai mencocokkan karakter dari sebelah kanan *pattern* (pola yang dicari). Algoritma *Brute Force* adalah algoritma untuk mencocokkan *pattern* dengan semua *teks* antara 0 dan n-m untuk menemukan keberadaan *pattern* dalam *teks*. Maka perlu adanya analisa yang membandingkan algoritma *Boyer Moore* dan *Brute Force* pada pencarian Kamus Besar Bahasa Indonesia berbasis android. Hasil dari penelitian ini menunjukkan bahwa algoritma *Brute Force* lebih cepat dibandingkan algoritma *Boyer Moore* pada pencarian Kamus Besar Bahasa Indonesia, Hasil total *running time* algoritma *Brute Force* adalah 168,3 ms pada kata, 6994,16ms pada deskripsi dan algoritma *Boyer Moore* adalah 304,7 ms pada kata, 8654,77 ms pada deskripsi. Untuk pengujian *keyword related* kedua algoritma dapat menampilkan daftar *keyword related* yang sama. Pada algoritma *Boyer Moore* maupun *Brute Force*, semakin sedikit karakter *pattern* yang dicari maka pencariannya semakin lama karena 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.

**Kata kunci :** *String Matching, Algoritma Boyer Moore, Algoritma Brute Force, Kamus Besar Bahasa Indonesia, Android.*

## I Pendahuluan

### 1. Latar Belakang

*String matching* adalah suatu algoritma yang digunakan untuk memecahkan masalah pencocokan suatu *teks* terhadap suatu *teks* lain. Ada beberapa algoritma yang dapat digunakan dalam pencarian *string matching* contohnya ialah algoritma *Boyer Moore* dan *Brute Force*.

Algoritma *Boyer-Moore* adalah salah satu algoritma pencarian *string*, dipublikasikan oleh Robert S. Boyer, dan J. Strother Moore pada tahun 1977. Algoritma ini dianggap sebagai algoritma yang paling efisien pada aplikasi umum. Tidak seperti algoritma pencarian *string* yang ditemukan sebelumnya, algoritma *Boyer-Moore* mulai mencocokkan karakter dari sebelah kanan *pattern* (pola yang dicari). Ide dibalik algoritma ini adalah bahwa dengan memulai pencocokkan karakter dari kanan, dan bukan dari kiri, maka akan lebih banyak informasi yang didapat. (Helmi, 2013)

Algoritma *Brute Force* adalah algoritma untuk mencocokkan *pattern* dengan semua *teks* antara 0 dan n-m untuk menemukan keberadaan *pattern* dalam *teks* (Riyanarto Sarno, Yeni Anistyasari, dan Rahimi Fitri, 2012). Di dalam pencocokkan *string*, terdapat

istilah *teks* dan *pattern*. *Teks* merupakan kata yang dicari dan dicocokkan dengan *pattern*. Sedangkan *pattern* merupakan kata yang diinputkan untuk dicocokkan.

Dari penelitian sebelumnya dilakukan analisa perbandingan pada algoritma *Boyer Moore* dan *Brute force* dalam pencarian *word suggestion* menggunakan Metode Perbandingan Eksponensial kemudian didapatkan kesimpulan bahwa algoritma *Boyer Moore* dan *Brute Force* dapat digunakan dalam pencarian *word suggestion*. (Andri Januardi, 2013). dan pada penelitian Sejenis dengan membandingkan algoritma *Boyer Moore* dan *Knuth Morris Path* dalam pencarian judul buku diketahui bahwa algoritma *Boyer Moore* lebih cepat dari pada *Knuth Morris path* dalam melakukan pencarian. (Guidio Leonardo Ginting, 2017).

Algoritma *Boyer Moore* dan *Brute Force* memiliki cara kerja yang berbeda, yaitu *Boyer Moore* melakukan pencocokan kata dimulai dari posisi kanan, sedangkan *Brute Force* melakukan pencocokan kata dimulai dari kiri sehingga jika diterapkan dalam pencarian Kamus Besar Bahasa Indonesia akan membuat kecepatan dalam pencariannya berbeda. Dengan adanya perbedaan tersebut, maka perlu adanya analisa yang membandingkan algoritma *Boyer Moore* dan *Brute Force*

## 2. Rumusan Masalah

Adapun rumusan masalah yang dihadapi dalam Tugas Akhir ini adalah :

- a. Bagaimana mengukur kinerja Algoritma *Booyer Moore* dan Algoritma *Brute Force* pada pencarian Kamus Besar Bahasa Indonesia ?
- b. Manakah algoritma yang lebih efisien untuk pencarian Kamus Besar Bahasa Indonesia ?

## 3. Batasan Masalah

Adapun batasan masalah yang dihadapi dalam Tugas Akhir ini adalah :

- a. Aplikasi berbasis *android* .
- b. Manajemen Sistem *Database* yang digunakan dalam program kamus berbasis android adalah *MySQL*.
- c. Sumber data didapat dari Kamus Besar Bahasa Indonesia.
- d. Jumlah kata yang tersedia dalam *database* berjumlah 17000 kata pada Kamus Besar Bahasa Indonesia.
- e. Parameter yang digunakan dalam perbandingan Algoritma *Boyer Moore* dan *Brute Force* adalah *Running Time* dan *Keyword related*.

## 4. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut:

- a. Mengukur kinerja Algoritma *Booyer Moore* dan Algoritma *Brute Force* pada pencarian Kamus Besar Bahasa Indonesia.

- b. Menemukan algoritma yang lebih efisien antara *algoritma Boyer Moore* dan *Brute Force* untuk pencarian kamus besar bahasa indonesia.

## 5. Manfaat Penelitian

Manfaat yang didapat dari Tugas Akhir ini adalah :

- a. Menambah pengetahuan dalam merancang aplikasi berbasis android dan pengetahuan tentang algoritma *Boyer Moore* dan Algoritma *Brute Force*.
- b. Manfaat ilmiah yang didapat adalah memberi pengetahuan mengenai kinerja algoritma *Boyer Moore* dan *Brute Force* serta memberi pengetahuan algoritma yang lebih efisien diantara kedua algoritma tersebut.
- c. Manfaat bagi masyarakat adalah agar pemilik *Smartphone* berbasis android dapat menggunakan aplikasi Kamus Besar Bahasa Indonesia untuk mempermudah dalam mengetahui kata dalam Kamus Besar Bahasa Indonesia.

## II. Tinjauan Pustaka

### 1. Algoritma *Boyer-Moore*

Algoritma *Boyer-Moore* adalah salah satu algoritma pencarian *string*, dipublikasikan oleh Robert S. Boyer, dan J. Strother Moore pada tahun 1977. Algoritma ini dianggap sebagai algoritma yang paling efisien pada aplikasi umum. Tidak seperti algoritma pencarian *string*

yang ditemukan sebelumnya, algoritma *Boyer-Moore* mulai mencocokkan karakter dari sebelah kanan *pattern* (pola yang di cari). Ide dibalik algoritma ini adalah bahwa dengan memulai pencocokkan karakter dari kanan, dan bukan dari kiri, maka akan lebih banyak informasi yang didapat. (Helmi, 2013).

## 2. Kelebihan dan kelemahan Algoritma Boyer Moore

*Boyer-Moore* membandingkan karakter dari kanan ke kiri dan memiliki loncatan karakter yang besar sehingga mempercepat pencarian *string* karena dengan hanya memeriksa sedikit karakter, dapat langsung diketahui bahwa *string* yang dicari tidak ditemukan dan dapat digeser ke posisi berikutnya. kelemahan algoritma *Boyer-Moore* mencocokkan *pattern* dari kanan ke kiri oleh sebab itu kelemahan dari algoritma ini adalah ketika semua karakter memiliki kesamaan atau cocok dan hanya karakter terakhir atau karakter paling kiri yang berbeda maka pencarian ini akan memerlukan waktu yang sedikit lama (Utomo, 2008).

Tabel 2.1. Contoh algoritma Boyer-Moore

<b>Teks</b>	G	R	A	C	E			
<b>Pattern</b>	H	A	L	I	M			

Pada tabel 2.1, dengan melakukan pencocokkan dari posisi paling akhir/kanan *pattern* dapat dilihat bahwa karakter “M” pada *pattern* “HALIM” tidak cocok dengan karakter “E” pada *teks* “GRACE”, dan karakter “E” tidak pernah ada dalam *pattern* “HALIM” yang dicari

sehingga *pattern* “HALIM” dapat digeser melewati *teks* “GRACE” sehingga posisinya menjadi:

Tabel 2.2. Contoh pergeseran algoritma Boyer-Moore

<b>Teks</b>	G	R	A	C	E						
<b>Pattern</b>							H	A	L	I	M

Tabel 2.2 menunjukkan bahwa algoritma *Boyer-Moore* memiliki pergeseran karakter yang besar sehingga mempercepat pencarian *pattern* karena dengan hanya memeriksa sedikit karakter, dapat langsung diketahui bahwa *pattern* yang dicari tidak ditemukan dan dapat digeser ke posisi berikutnya (Ginting, 2014)

Algoritma *Boyer-Moore* menggunakan dua buah tabel untuk mengolah informasi saat terjadi kegagalan pencocokan *pattern*. Tabel pertama disebut *bad character shift* juga sering disebut *occurrence heuristic* (OH). Tabel kedua disebut dengan istilah *good suffix shift* juga disebut *match heuristic* (MH) (Charras, 2014)

Secara sistematis, langkah-langkah yang dilakukan algoritma *Boyer-Moore* pada saat mencocokkan *pattern* adalah:

1. Algoritma *boyer-moore* mulai mencocokkan *pattern* pada karakter paling akhir/kanan.
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 mengambil nilai terbesar dari penggeseran *good-suffix* dan penggeseran *bad-character*, lalu mengulangi langkah 2 sampai *pattern* berada di ujung *teks*. (Ramadhansyah,2013)

### 3. Algoritma Brute Force

Algoritma *Brute Force* adalah algoritma untuk mencocokkan *pattern* dengan semua *teks* antara 0 dan n-m untuk menemukan keberadaan *pattern* dalam *teks* (Riyanarto Sarno, Yeni Anistiyasari, dan Rahimi Fitri, 2012). Di dalam pencocokkan *string*, terdapat istilah *teks* dan *pattern*. *Teks* merupakan kata yang dicari dan dicocokkan dengan *pattern*. Sedangkan *pattern* merupakan kata yang diinputkan untuk dicocokkan. Secara rinci, langkah – langkah yang dilakukan algoritma ini saat mencocokkan *string* adalah:

1. Algoritma *Brute Force* mulai mencocokkan *pattern* dari awal *teks*.
2. Dari kiri ke kanan, algoritma ini akan mencocokkan karakter per karakter *pattern* dengan karakter pada *teks* yang

bersesuaian, sampai salah satu kondisi berikut terpenuhi :

- a. Karakter di *pattern* dan di *teks* yang dibandingkan tidak cocok.
  - 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*.
4. Kelemahan dan kelebihan algoritma *Brute Force*

Algoritma *Brute Force* juga memiliki kelebihan dan kelemahan. Adapun kelebihan dari algoritma *Brute Force* yaitu:

- a. Algoritma *Brute Force* dapat digunakan untuk memecahkan hampir sebagian besar masalah.
- b. Algoritma *Brute Force* sederhana dan mudah dimengerti.
- c. Algoritma *Brute Force* menghasilkan algoritma yang layak untuk beberapa masalah penting seperti pencarian, pengurutan, pencocokkan *string* , atau perkalian *matriks*.
- d. Algoritma *Brute Force* menghasilkan algoritma baku (*standard*) untuk tugas- tugas komputasi penjumlahan / perkalian n buah bilangan, menentukan elemen

minimum atau maksimum di dalam tabel (*list*).

Sedangkan kelemahan dari algoritma *Brute Force* yaitu sebagai berikut:

- a. *Algoritma Brute Force* jarang menghasilkan algoritma yang manjur.
- b. Beberapa algoritma *Brute Force* lambat, sehingga tidak dapat diterima.
- c. Tidak sekonstruktif/sekreatif teknik pemecahan masalah lainnya.

## 5. Kamus

Kamus adalah sejenis buku rujukan yang menerangkan makna kata-kata. Kamus berfungsi untuk membantu seseorang mengenal perkataan baru. Selain menerangkan maksud kata kamus juga mungkin mempunyai pedoman sebutan, asal usul (etimologi) sesuatu perkataan dan juga contoh penggunaan bagi sesuatu perkataan. Untuk memperjelas kadang kala terdapat juga ilustrasi didalam kamus (Susanto,2014).

## 6. Kamus Besar Bahasa Indonesia

Kamus bahasa Indonesia terbitan Pusat Bahasa pertama adalah Kamus Umum Bahasa Indonesia (1952) yang diselenggarakan oleh W.J.S. Poerwadarminta. Edisi kelima terbit pada tahun 1976. Kemudian pada tahun 1988 terbit Kamus Besar Bahasa Indonesia

yang dimaksudkan sebagai kamus baku untuk bahasa Indonesia. Kamus ini merupakan hasil karya tim, dengan pemimpin redaksi Sri Sukei Adiwimarta 10 dan Adi Sunaryo, dan penyelia Anton M. Moeliono. Edisi ketiga Kamus Besar Bahasa Indonesia diterbitkan pada tahun 2002. Kamus edisi ketiga ini memuat sekitar 78.000 lema.( UPN "Veteran" Jatim)

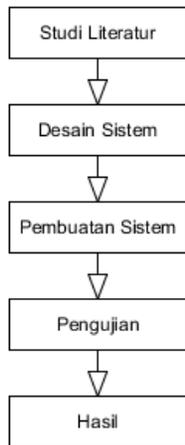
Berikut beberapa contoh kata dari KBBI :

1. **aba-aba** n kata-kata perintah atau komando, spt dl baris-berbaris, senam, atau menyanyi bersama, msl siap!, kiri! kanan!, satu! dua!, maju jalan!
2. **bahas, membahas** v mempertemukan atau memasang papan dng memakai baji atau gandar.
3. **bakpao** n kue spt roti terbuat dr terigu dng isi daging atau kacang hijau.
4. **lancang** n mangkuk kecil yg bertangkai (tempat air teh atau kopi yg hendak diminum).

## III Metode Penelitian

### 1. Tahapan Penelitian

Dalam pengerjaan Tugas Akhir ini diperlukan langkah – langkah penelitian yang mendukung dan memaksimalkan dalam penyelesaiannya. Pada metode penelitian terdapat langkah – langkah sebagai berikut :



Keterangan Tahapan Penelitian:

1. Studi literatur  
Pengumpulan data dengan cara mengumpulkan jurnal, paper, literatur, yang terkait dengan judul penelitian, guna melengkapi pengetahuan dasar, mempelajari dan memahami Algoritma *Boyer Moore* dan *Brute Force*.
2. Desain Sistem  
Mendesain sistem Aplikasi Kamus Besar Bahasa Indonesia dengan kedua algoritma tersebut.
3. Pembuatan Sistem  
Membuat sistem Aplikasi KBBI menggunakan algoritma *Boyer Moore* dan *Brute Force*.
4. Pengujian  
Melakukan pengujian apakah aplikasi dapat berfungsi dengan baik, kemudian melakukan

analisis perbandingan kedua algoritma tersebut.

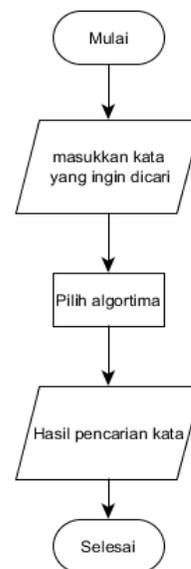
#### 5. Hasil

Pada tahap ini dilakukan pembuatan laporan yang disusun berdasarkan hasil penelitian sehingga menjadi laporan penelitian yang dapat memberikan gambaran secara utuh tentang sistem yang sedang dibangun.

#### 2. Flowchart

*Flowchart* merupakan gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta pernyataannya. Gambaran ini dinyatakan dengan simbol. Dengan demikian setiap simbol menggambarkan proses tertentu. Sedangkan antara proses digambarkan dengan garis penghubung. *Flowchart* yang akan dibangun dapat ditunjukkan pada **Gambar 3.2**, **3.3**, dan **3.4**

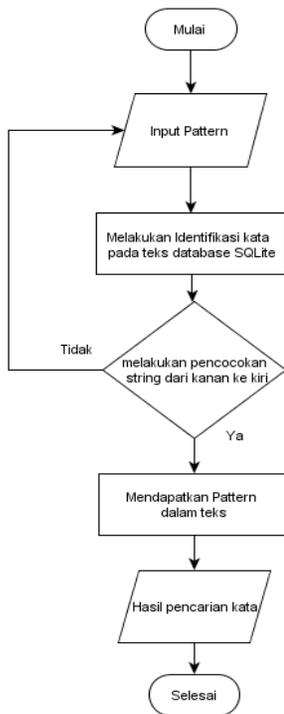
##### a. *Flowchart* Gambaran Umum Sistem



Gambar 3.2 *Flowchart* gambaran umum sistem

Gambar 3.2 Menggambarkan alur sistem secara umum pada kamus besar bahasa Indonesia, dimana *user* menginput kata yang ingin dicari. Selanjutnya *user* memilih Algoritma untuk pencarian kata. Maka sistem akan mencocokkan kata sesuai dengan Algoritma yang telah dipilih lalu sistem akan menampilkan hasil pencarian kata.

b. *Flowchart Proses Boyer-Moore*



Gambar 3.3 *Flowchart* proses *Boyer Moore*

Dari Gambar 3.3 dapat dilihat proses *Boyer-Moore* yang terdapat pada sistem yang akan dibangun ini. Pertama masukkan *pattern* berupa karakter untuk dapat melakukan pencarian *string*. Kemudian melakukan pembacaan *pattern* pada *teks* pada *SQL*, apakah *string* dicari merupakan cocok yang terdapat pada *teks*. Melakukan pencocokan *string*

dari kana ke kiri sesuai dengan inputan *pattern* yang diinginkan oleh sipengguna. Dan dilakukan sampai *pattern* yang dicari dilakukan pencocokan. Setelah melakukan pencocokan, maka *pattern* akan mendapatkan kecocokan pada *teks* yang telah tersedia. Dimana sudah terjadi pengkelompokan data untuk langsung dapat mengetahui artinya. Akhirnya sistem mengeluarkan *output*, dimana sistem ini secara otomatis mengeluarkan semua pilihan berdasarkan *pattern* yang diinputkan.

c. *Flowchart Proses Brute Force*



Gambar 3.3 *Flowchart* proses *Brute Force*

Dari **Gambar 3.11** dapat dilihat proses *Brute Force* yang terdapat pada sistem yang akan dibangun ini. Pertama masukkan *pattern* berupa karakter untuk dapat melakukan pencarian *string*. Kemudian melakukan pembacaan *pattern* pada *teks* pada *SQL*, apakah *string* dicari merupakan cocok yang terdapat pada *teks*. Melakukan pencocokan *string* dari kiri ke kanan sesuai dengan inputan *pattern* yang diinginkan oleh si pengguna. Dan dilakukan sampai *pattern* yang dicari dilakukan pencocokan. Setelah melakukan pencocokan, maka *pattern* akan mendapatkan kecocokan pada *teks* yang telah tersedia. Dimana sudah terjadi pengkelompokan data untuk langsung dapat mengetahui artinya. Akhirnya sistem mengeluarkan *output*, dimana sistem ini secara otomatis mengeluarkan semua pilihan berdasarkan *pattern* yang diinputkan dan menampilkan *running time* proses pencarian menggunakan algoritma *Brute Force*.

#### IV Implementasi dan Pengujian

##### 1. Pengujian Sistem

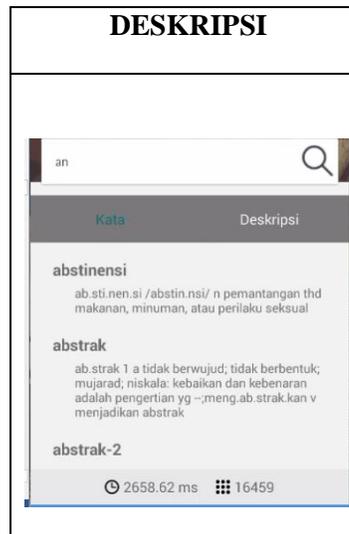
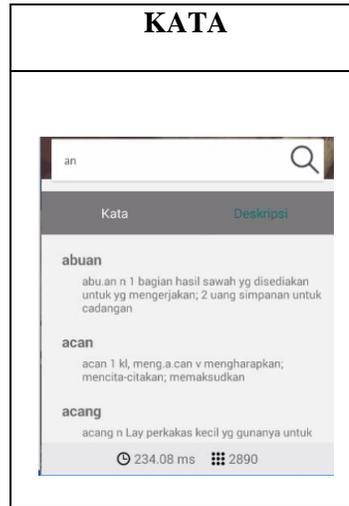
Pengujian sistem dilakukan untuk mengetahui bagaimana kinerja sistem dalam pencarian kata pada Kamus Besar Bahasa Indonesia menggunakan algoritma *Boyer Moore* dan *Brute Force*. Dalam pengujian yang akan dicari adalah *string* sebagai inputannya. Semua hasil pencarian ditampilkan sesuai dengan inputan pencarian. Jumlah kata pada *database* untuk pengujian sistem berjumlah 17000 kata pada Kamus Besar Bahasa Indonesia dengan menggunakan parameter pengujian *running time* dan *keyword related*.

##### 2. Pengujian Pencarian kata pada Kamus Besar Bahasa Indonesia dengan algoritma *Boyer Moore*.

Pada pengujian kata Kamus Besar Bahasa Indonesia dengan algoritma *Boyer Moore* dilakukan 20 kali pengujian kata dengan beberapa karakter yang digunakan sebagai

sampel pengujian, dapat dilihat pada tabel 4.1 sebagai contoh pengujian.

**Tabel 4.1** melakukan pengujian kata “an” dengan algoritma *Boyer Moore*



Pada tabel 4.1 menjelaskan tentang hasil pencarian kata “an” pada kata dan deskripsi dengan waktu pencarian 234,08 ms pada kata dan 2658,62 ms pada deskripsi dan jumlah kata yang ditemukan kata = 2890, deskripsi = 164959.

Untuk pengujian ke-2 sampai ke-20 dapat dilihat pada tabel 4.2 dibawah ini.

**Tabel 4.2** Pengujian kata dengan algoritma *Boyer Moore*

<i>Boyer moore</i>					
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
Total		304,7	8654,77	4705	45672

Pada tabel 4.3 dapat dijeleaskan dari hasil pengujian kata menggunakan algoritma *Boyer Moore* dengan 20 kali percobaan diperoleh total *running time* kata = 305,59 ms ,deskripsi = 8722,26 ms dan jumlah kata yang ditemukan kata = 4705, deskripsi = 45662.

3. Pengujian Pencarian kata pada Kamus Besar Bahasa Indonesia dengan algoritma *Brute Force*.  
 Pada pengujian kata Kamus Besar Bahasa Indonesia dengan algoritma *Brute Force* dilakukan 20 kali pengujian kata dengan beberapa karakter yang digunakan sebagai sampel pengujian, dapat dilihat pada tabel 4.3.

**Tabel 4.3** Pengujian kata 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
Total		168,3	6994,16	4705	45672

Pada tabel 4.3 dapat dijelaskan dari hasil pengujian kata menggunakan algoritma *Brute Force* dengan 20 kali percobaan diperoleh total *running time* kata = 168,38 ms ,deskripsi = 7097,64 ms dan jumlah kata yang ditemukan kata = 4705, deskripsi = 45662.

4. Pengujian Pencarian kata palindrom pada Kamus Besar Bahasa Indonesia dengan algoritma *Boyer moore* dan *Brute Force*  
 Pada pengujian kata palindrom atau kata yang dapat dibaca dengan sama baik dari depan maupun belakang pada Kamus Besar Bahasa Indonesia dengan algoritma *Boyer Moore* dan *Brute Force* dilakukan 5 kali pengujian kata dengan beberapa karakter yang digunakan sebagai sampel

pengujian, dapat dilihat pada tabel 4.4 sebagai pengujian.

Tabel 4.4 Pengujian kata palindrom dengan algoritma *Boyer moore* dan *Brute Force*

No	Pattern	Running Time (ms)				Keyword related	
		Boyer moore		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	makan	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>		9,67	1251,6	6,42	1068,47	266	8599
<b>Rata-rata</b>		1,93	250,32	1,28	213,69		

Pada tabel 4.4 dapat dijelaskan dari hasil pengujian kata palindrom menggunakan algoritma *Boyer Moore* dan *Brute Force* dengan 5 kali percobaan diperoleh rata-rata *running time Boyer Moore* kata = 1,93 ms, deskripsi = 250,32 ms, *Brute Force* kata = 1,28 ms, deskripsi = 213,69 ms dan jumlah kata yang ditemukan kata = 266, deskripsi = 8599.

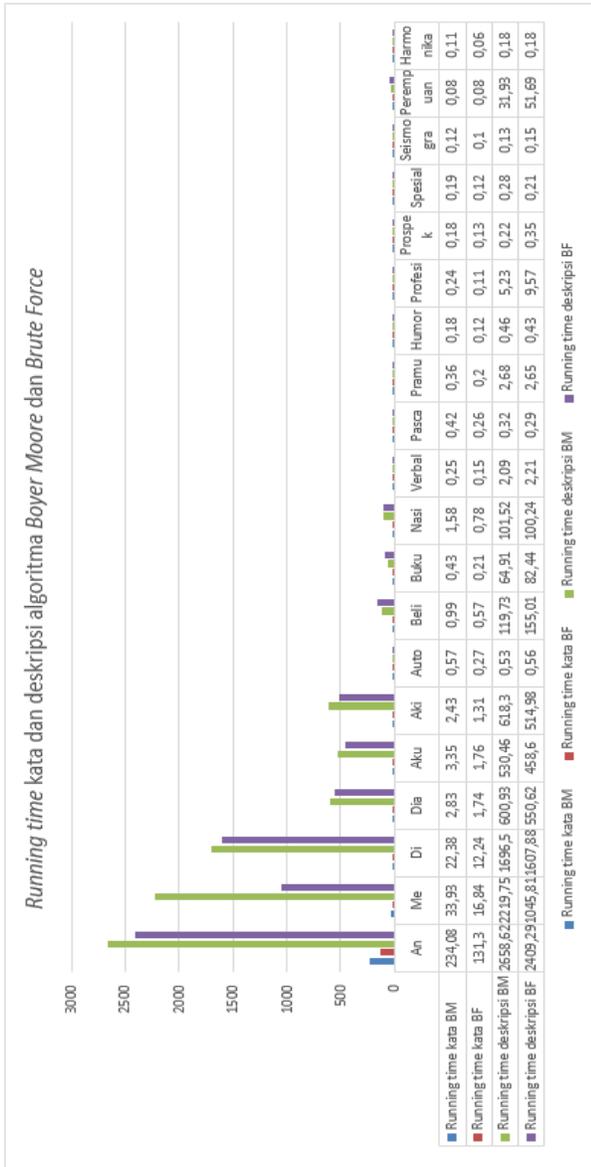
### 5. Pengujian *Running Time* dan *Keyword Related*

Pengujian *running time* dilakukan untuk mengukur kecepatan algoritma saat melakukan pencarian kata. Hasil pengujian *running time* kata dan deskripsi dari kedua algoritma yang digunakan akan dijelaskan pada tabel 4.4.

Tabel 4.5 Hasil pengujian *running time* dan *keyword related* algoritma *Boyer Moore* dan *Brute Force*

NO	Pattern	Boyer moore		Brute Force	
		Running Time (ms)		Running Time (ms)	
		Kata	Deskripsi	Kata	Deskripsi
1	An	234,08	2658,62	131,3	2409,29
2	Me	33,93	2219,75	16,84	1045,81
3	Di	22,38	1696,5	12,24	1607,88
4	Dia	2,83	600,93	1,74	550,62
5	Aku	3,35	530,46	1,76	458,6
6	Aki	2,43	618,3	1,31	514,98
7	Auto	0,57	0,53	0,27	0,56
8	Beli	0,99	119,73	0,57	155,01
9	Buku	0,43	64,91	0,21	82,44
10	Nasi	1,58	101,52	0,78	100,24
11	Verbal	0,25	2,09	0,15	2,21
12	Pasca	0,42	0,32	0,26	0,29
13	Pramu	0,36	2,68	0,2	2,65
14	Humor	0,18	0,46	0,12	0,43
15	Profesi	0,24	5,23	0,11	9,57
16	Prospek	0,18	0,22	0,13	0,35
17	Spesial	0,19	0,28	0,12	0,21
18	Seismogra	0,12	0,13	0,1	0,15
19	Perempuan	0,08	31,93	0,08	51,69
20	Harmonika	0,11	0,18	0,06	0,18
<b>Total</b>		304,7	8654,77	168,3	6994,16
<b>Rata-rata</b>		15,235	432,7	8,415	349,7

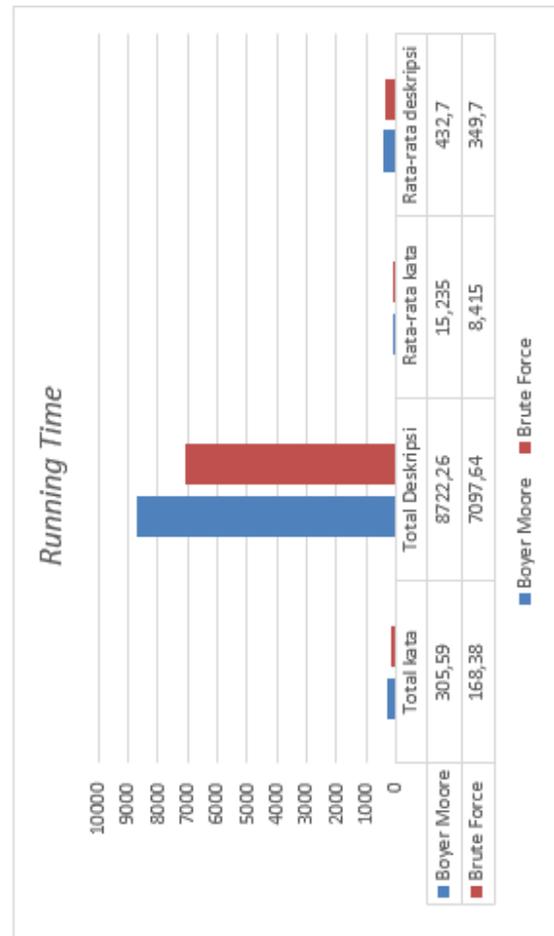
Setelah mendapatkan hasil pengujian dari tabel 4.4 maka dibuat grafik perbandingan hasil pengujian dari kedua algoritma tersebut. Grafik dapat dilihat pada **gambar 4.1**



**Gambar 4.1** Perbandingan hasil *running time* kata dan deskripsi algoritma *Boyer Moore* dan *Brute Force*

Bedasarkan grafik diatas dapat dijelaskan bahwa algoritma *Brute Force* mendapatkan hasil *running time* yang relatif rendah dibandingkan dengan algoritma *Boyer moore*. Artinya bahwa algoritma *Brute Force* lebih cepat untuk pencocokan kata pada Kamus Besar Bahasa Indonesia dibandingkan dengan algoritma *Boyer Moore*. Total hasil perbandingan dari kedua algoritma tersebut dapat dilihat pada **gambar 4.2**

**Gambar 4.2** Perbandingan total *running time* algoritma *Boyer Moore* dan algoritma *Brute Force*



Dari grafik diatas dapat dijelaskan bahwa algoritma *Brute Force* memiliki nilai total dan rata-rata *running time* yang rendah dibandingkan

dengan algoritma *Boyer Moore*. artinya bahwa algoritma *Brute Force* lebih cepat untuk pencocokan kata pada Kamus Besar Bahasa Indonesia dibandingkan dengan algoritma *Boyer Moore*.

Berdasarkan gambar 4.7 diperoleh rata-rata *Boyer moore*, kata = 15,235, deskripsi 432,7, *Brute Force*, kata = 8,415 deskripsi = 349,7 Jika dihitung prosentasenya, akan diperoleh.

$$\text{Kata} = 15,235 : 8,415 = 3 : 2$$

$$= \frac{2}{3} \times 100\% = 66,6\%$$

$$\text{Deskripsi} = 432,7 : 349,7 = 9 : 7$$

$$= \frac{7}{9} \times 100\% = 77,7\%$$

Jadi dapat diketahui nilai prosentase dari perbandingan kedua algoritma bahwa algoritma *Brute Force* lebih cepat 66,6 % pada kata dan 77,7% lebih cepat pada deskripsi dibandingkan *Boyer moore*.

## 6. Analisa Pengujian

Analisa pengujian dilakukan untuk menganalisa data berdasarkan data yang diperoleh dari pengujian hasil pengujian *running time* dan *keyword related* algoritma *Boyer Moore* dan *Brute Force*, diperoleh analisa pengujian.

### a. Analisa running time

Pada pengujian pencarian kata pada Kamus Besar Bahasa Indonesia algoritma *Brute Force* memiliki *running time* yang relatif rendah dibandingkan *Boyer Moore*, dikarenakan algoritma *Brute Force* melakukan pencocokan

dimulai dari kiri ke kanan dengan lompatan pencarian satu persatu dan apabila kata yang dicari ketemu maka pencarian berhenti, sedangkan algoritma *Boyer Moore* melakukan pencocokan dimulai dari kanan ke kiri dengan lompatan sejumlah karakter yang dicari, apabila kata yang dicari ketemu maka pencarian akan terus berlanjut sampai karakter paling kanan pada *pattern* tidak memiliki pasangan yang sejajar untuk dibandingkan. dengan jumlah data 17000 kata maka algoritma *Brute Force* lebih cepat dibandingkan *Boyer Moore*. Pada pengujian pencarian kata palindrom pada Kamus Besar Bahasa Indonesia algoritma *Brute Force* memiliki *running time* yang relatif rendah dibandingkan *Boyer Moore*.

Pada algoritma *Boyer Moore* maupun *Brute Force*, semakin sedikit karakter *pattern* yang dicari maka pencariannya semakin lama karena yang dicari semakin banyak dan kata yang ditemukan juga semakin banyak, dan semakin banyak karakter *pattern* yang dicari maka pencariannya semakin lama karena yang dicari semakin sedikit dan kata yang ditemukan juga semakin sedikit.

### b. Analisa keyword related

Untuk pengujian *keyword related* kedua algoritma dapat menemukan semua kata yang dicari dengan jumlah yang sama, artinya kedua algoritma dapat menampilkan daftar *keyword related* yang sama.

## V Penutup

### A. Kesimpulan

Berdasarkan hasil penelitian, analisis, perancangan, implementasi, dan pengujian sistem ini, maka kesimpulan yang didapat adalah sebagai berikut:

1. Pada pengujian *running time* dapat disimpulkan bahwa algoritma *Brute Force* memiliki nilai total dan rata-rata yang rendah dengan prosentase 66,6 % pada kata dan 77,7% pada deskripsi lebih cepat untuk pencocokan kata pada Kamus Besar Bahasa Indonesia dibandingkan dengan algoritma *Boyer Moore*.
2. Pada pengujian pencarian kata palindrom dapat disimpulkan bahwa algoritma *Brute Force* memiliki nilai total dan rata-rata yang rendah dibandingkan dengan algoritma *Boyer Moore*. Artinya bahwa algoritma *Brute Force* lebih cepat pada pencarian kata palindrom dibandingkan dengan algoritma *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. Untuk *keyword related* kedua algoritma dapat menemukan semua kata yang dicari dengan jumlah yang sama, artinya kedua algoritma dapat digunakan untuk menampilkan *keyword related*.

## **B. Saran**

Adapun saran-saran yang diperlukan untuk penelitian maupun pengembangan berikutnya adalah:

1. Dalam pengembangan aplikasi berikutnya diharapkan aplikasi ini menyediakan menu pilihan algoritma pencarian string yang baru agar terlihat perbedaan pada pencarian string antara algoritma yang lama dengan algoritma yang sudah mengalami proses perkembangan.
2. Dengan jumlah data 17000 kata pada *database* Kamus Besar Bahasa Indonesia yang digunakan pada pengujian saat ini, diharapkan pada penelitian algoritma *Boyer Moore* dan *Brute Force* selanjutnya jumlah data yang digunakan untuk pengujian lebih banyak dari jumlah data yang ada pada penelitian ini.
3. Diharapkan Algoritma *Boyer moore* dan *Brute Force* bisa digunakan untuk studi kasus yang lain.

## **Daftar Pustaka**

- Januardi, Andri. 2013. Analisa Perbandingan Algoritma Brute Force Dan Boyer Moore Dalam Pencarian Word Suggestion Menggunakan Metode Perbandingan Eksponensial. Pelita Informatika Budi Darma IV(4).
- Fau, Alwin, Mesran dan Ginting Guidio Leonarde. (2017) Analisa Perbandingan Algoritma Boyer Moore dan Knuth

Morris Pratt Dalam Pencarian Judul Buku  
Menerapkan Metode Perbandingan  
Ekspensial. STMIK Budi Darma.

thesis, Universitas Kristen Maranatha.  
(3).

Ruslan, Imran, Nur. Pramono Bambang dan  
Subardin. (2016). APLIKASI KAMUS  
BAHASA LATIN HEWAN DAN  
TUMBUHAN MENGGUNAKAN METODE  
BRUTE FORCE DAN FITUR  
AUTOCOMPLETE BERBASIS  
ANDROID. Universitas Halu  
Oleo, Kendari

Kencana Wulan Argakusumah dan Seng Hansun.  
(2016) Implementasi Algoritma *Boyer-  
Moore* pada Aplikasi Kamus  
Kedokteran Berbasis Android. Universitas  
Multimedia Nusantara,  
Tangerang.

Gunawan. (2016) APLIKASI KAMUS ISTILAH  
EKONOMI (INGGRIS-INDONESIA)  
MENGGUNAKAN METODE  
SEQUENTIAL SEARCHING. Universitas  
Muhammadiyah Bengkulu.

Ramadhansyah. 2013. Perancangan Aplikasi  
Kamus Bahasa Gayo dengan Menggunakan  
Metode Boyer-Moore. Jurnal Pelita  
Informatika Budi Darma IV.

Lukito, Lanvin. 2012. Studi  
Perbandingan Algoritma Brute Force,  
Boyer Moore dan Deterministic  
Finite Automata dalam Pencarian String  
pada Dokumen. Undergraduate