

KOMBINASI ALGORITMA *BRUTE FORCE* DAN *STEMMING* PADA SISTEM PENCARIAN *MASHDAR*

Rozzi Kesuma Dinata¹, Safwandi², Novia Hasdyna³, Rudi Mahendra⁴

^{1,2,4} Program Studi Teknik Informatika Universitas Malikussaleh

Jl. Kampus Unimal Bukit Indah, Blang Pulo, Lhokseumawe, Aceh, Indonesia

³ Program Studi Teknik Informatika Universitas Islam Kebangsaan Indonesia

Jl. Medan – Banda Aceh Sp. Alue Awe, Lhokseumawe, Aceh, Indonesia

¹rozzi@unimal.ac.id, ²safwandi@unimal.ac.id, ³noviahasdyna@gmail.com, ⁴rudimahend7@gmail.com

Abstrak—Penelitian ini mengkombinasikan algoritma *Brute Force dan Stemming* pada sistem pencarian kata *mashdar*. Tujuan penelitian ini adalah untuk memudahkan pengguna dalam mencari kata dasar dari suatu kata beserta dengan perubahannya dengan berdasarkan tinjauan kata dasar kepada *mashdar*. Adapun data yang digunakan dalam penelitian ini diperoleh dari kitab Tashrif karangan Syekh Hasan bin Ahmad sebanyak 50 data. Hasil penelitian dengan mengkombinasikan algoritma *brute force dan stemming* pada pencarian *mashdar* dengan *sample* pengujian sebanyak 15 kata maka diperoleh nilai akurasi sebanyak 78.9%, *precision* 86.6%, *recall* 84.6% dan *specificity* 66.6%.

Kata Kunci— Kombinasi, *Brute Force*, *Stemming*, *Mashdar*, Akurasi

Abstract—In this study *Brute Force and Stemming* algorithms are combined in the *mashdar* word search system. The purpose of this study is to facilitate the user in finding the basic words of a word along with its changes based on the review of basic words to *mashdar*. The data used in this study were obtained from 50 books of Tashrif by Sheikh Hasan bin Ahmad. The results of the study by combining *brute force and stemming* algorithms in *mashdar* search with a test sample of 15 words obtained an accuracy value of 78.9%, 86.6% *precision*, 84.6% *recall* and 66.6% *specificity*.

Keywords—Combination, *Brute Force*, *Stemming*, *Mashdar*, Accuracy

I. PENDAHULUAN

Kebutuhan terhadap informasi semakin meningkat, sehingga semakin banyak informasi yang dibutuhkan untuk dapat diperoleh dengan cepat, informasi tersebut baik berupa teks atau berbentuk gambar yang kemudian disajikan sebagai suatu bentuk data [1]. Suatu teks biasanya memiliki imbuhan baik *prefiks*, *infiks* dan *sufiks* yang berguna untuk melengkapi suatu kata, dalam bahasa Arab juga memiliki *afiks* seperti bahasa Indonesia yang berpengaruh terhadap perubahan makna dari kata tersebut. Maka perlu dilakukan pemotongan imbuhan dan mengembalikan suatu kata kepada kata dasar baik dengan tinjauan kata dasar dalam bentuk *fi'il madhi* ataupun *mashdar*, untuk lebih mudah memahami makna dari kata tersebut [2].

Salah satu algoritma yang digunakan untuk membantu teks mining adalah *stemming*[3]. Adapun proses *stemming* berfungsi untuk menemukan kata dasar dari sebuah kata dengan melakukan pemotongan imbuhan yang terdapat di awal (*prefiks*), di tengah (*infiks*) atau di akhir (*sufiks*) maka *stemming* digunakan untuk mengganti bentuk dari suatu kata

menjadi kata dasar dengan menghilangkan imbuhan yang terdapat pada teks, dalam teks bahasa Arab imbuhan ini tidak produktif karena pemakaiannya terbatas hanya pada kata-kata tertentu, pemakaian imbuhan hanya tertentu pada kalimat yang memiliki huruf *illat* (alif (ا), waw(و), ya(ي)) di dalam suku kata tersebut karena adanya perubahan bentuk dengan sebab *tashrif* maka kata tersebut harus dikembalikan kepada kata dasar semula untuk lebih mudah melakukan pencocokan dengan teks pada kamus di *database* [4].

Pada penelitian ini penulis mengkombinasikan algoritma *brute force* dan *stemming* pada sistem pencarian *mashdar*, untuk memudahkan masyarakat dalam mencari kata dasar dari suatu kata beserta dengan perubahannya dengan berdasarkan tinjauan kata dasar menjadi *mashdar*. Penelitian ini juga menganalisis nilai akurasi dari hasil kombinasi algoritma *brute force* dan *stemming*, berupa akurasi, *precision*, dan *recall*.

II. TINJAUAN PUSTAKA

A. Teks Mining

Dalam melakukan klasifikasi dokumen tekstual, biasanya digunakan *teks mining*. Dokumen-dokumen tersebut akan dilakukan proses pengklasifikasian sesuai dengan topic dokumen. Dengan teks mining suatu artikel dapat juga diketahui jenis kategorinya. Analisis dan pencocokan basis data kata kunci dengan kata-kata yang mewakili isi dari artikel. Adanya *teks mining* dapat membantu melakukan pengelompokkan suatu dokumen dengan waktu yang singkat [5].

B. Algoritma Brute Force

Sebuah konsep yang di dasarkan pada pernyataan masalah secara langsung (straightforward) yang digunakan sebagai solusi untuk pemecahan masalah pernyataan kalimat disebut dengan *Brute force*. Algoritma *brute force* memecahkan masalah dengan sangat sederhana, langsung dan dengan cara yang jelas (*obvious way*) [6].

C. Stemming

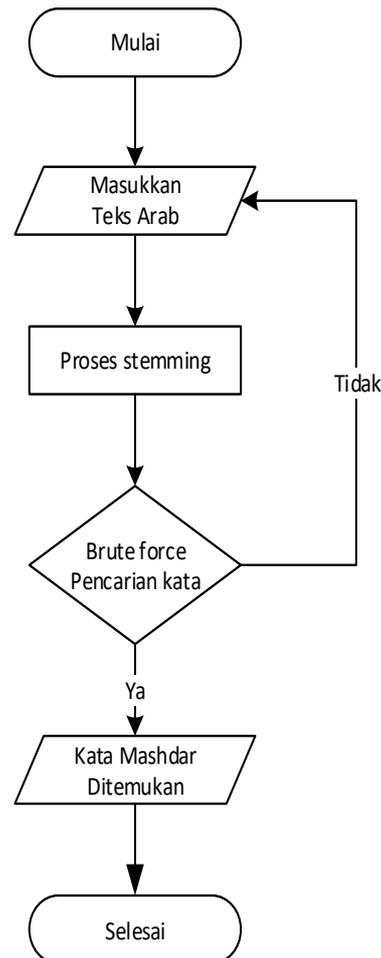
Stemming adalah suatu proses untuk menemukan kata dasar dari sebuah kata, dengan menghilangkan semua imbuhan (*affixes*) baik yang terdiri dari sisipan (*infixes*), awalan (*prefixes*), akhiran (*suffixes*) dan *confixes* (kombinasi dari awalan dan akhiran) pada kata turunan. Penggunaan *Stemming* biasanya digunakan untuk mengganti atau merangkai bentuk dari suatu kata menjadi kata dasar dari sebuah kata tersebut yang sesuai dengan struktur morfologi bahasa Indonesia yang baik dan benar. Imbuhan-imbuhan yang melekat pada suatu kata harus dihilangkan untuk mengubah bentuk kata tersebut menjadi bentuk kata dasarnya [7].

D. String matching

Proses pencarian semua kemunculan *query* yang selanjutnya disebut *pattern* ke dalam *string* yang lebih panjang atau teks merupakan *String Matching*. *String matching* dibagi menjadi dua, yakni *exact matching* dan *heuristic* atau *statistical matching*. Penggunaan *exact matching sendiri adalah untuk menemukan pattern yang berasal dari satu teks*. Contoh pencarian *exact matching* adalah pencarian kata “pelajar” dalam kalimat “saya seorang pelajar” atau “saya seorang siswa”. Sistem akan memberikan dan menampilkan hasil bahwa kalimat pertama mengandung kata “pelajar”, sedangkan kalimat kedua tidak, meskipun kenyataannya pelajar dan siswa adalah kata yang bersinonim [8].

III. METODOLOGI PENELITIAN

Adapun alur penelitian ini dapat dilihat pada diagram alir penelitian berikut.

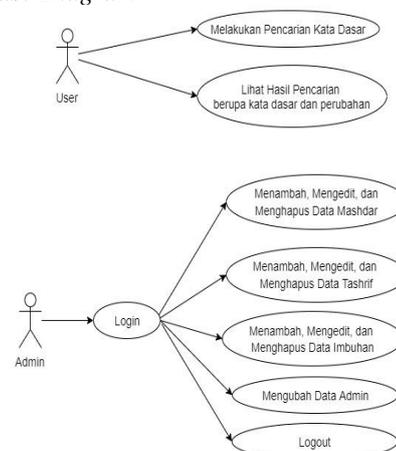


Gbr 1. Flowchart proses stemming dan brute force

IV. HASIL DAN PEMBAHASAN

A. Manajemen Basis Model

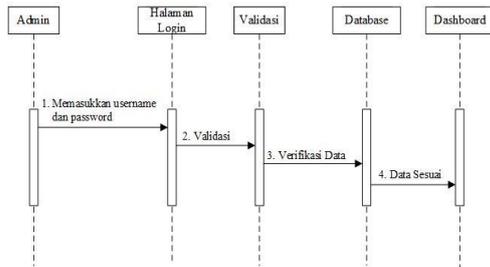
1. Use Case Diagram



Gbr 2. Use Case Diagram

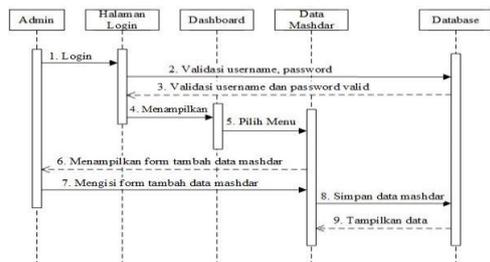
2. Sequence Diagram

a. Sequence diagram login



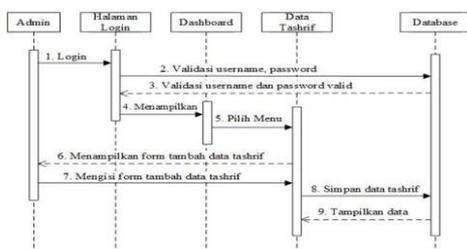
Gbr 3. sequence diagram login

b. Sequence diagram tambah mashdar



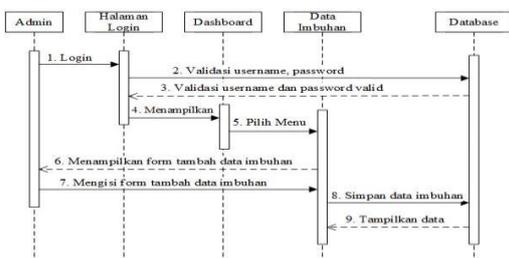
Gbr 4. Sequence diagram mashdar

c. Sequencediagram tambah tashrif



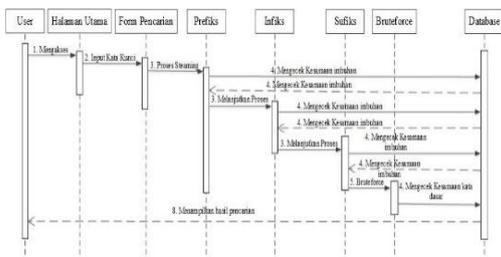
Gbr 5. Sequence diagram tashrif

d. Sequencediagram tambah imbuhan



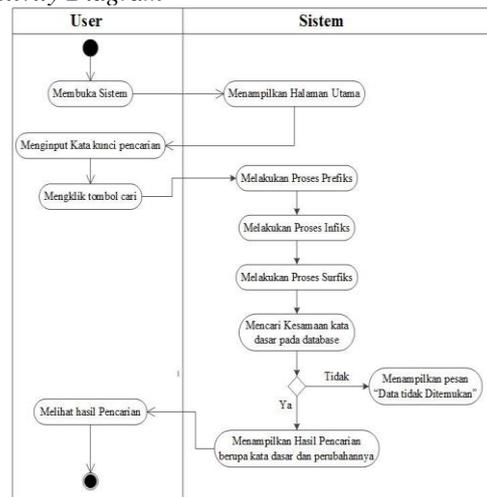
Gbr 6. Sequence diagram imbuhan

e. Sequencediagram proses



Gbr 7. Sequence diagram proses

3. Activity Diagram



Gbr 8. Activity diagram

B. ManajemenaBasis Data

1. Tabel_Admin

TABEL I
ADMIN

No	Nama	Type	Width	Keterangan
1	admin_id	Int	10	Primary key
2	username	Varchar	30	Username
3	password	Varchar	30	Password

2. Tabel_Mashdar

TABEL II
TABEL MASHDAR

No	Nama	Type	Width	Keterangan
1	Id_mashdar	Int	10	Primary key
2	mashdar	Varchar	30	Kata dasar

3. Tabel_Tashrif

TABEL III
TABEL TASHRIF

No	Nama	Type	Width	Keterangan
1	id_tsh	Int	10	Primary key
2	tsh_madhi	Varchar	50	Perubahan kata
3	tsh_mudhari	Varchar	50	Perubahan kata
4	tsh_mashdar	Varchar	50	Kata dasar
5	tsh_ism_fail	Varchar	50	Perubahan kata
6	tsh_ism_maful	Varchar	50	Perubahan kata
7	tsh_amr	Varchar	50	Perubahan kata
8	tsh_nahi	Varchar	50	Perubahan kata
9	tsh_ism_zaman	Varchar	50	Perubahan kata
10	tsh_ism_makan	Varchar	50	Perubahan kata
11	tsh_ism_alat	Varchar	50	Perubahan kata

4. Tabel_Imbuhan

TABEL V
TABEL IMBUHAN

No	Nama	Type	Width	Keterangan
1	Id_imbuhan	Int	10	Primary key
2	prefiks	Varchar	20	Imbuhan diawal
3	infiks	Varchar	20	Imbuhan ditengah
	sufiks	Varchar	20	Imbuhan diakhir

C. Proses Kombinasi Brute Force dan Stemming

Pada pengujian ini penulis menggambarkan bagaimana proses pemotongan dan pencarian kata *mashdar* dengan menampilkan *source code* program beserta dengan penjelasannya, sebagai berikut:

Page | 276

1. Proses pemotongan prefiks

```
function stemPrefiks($koneksi,$stemed='') {
    $ketemu=false;$hasil="";
    $result=mysqli_query($koneksi,"SELECT prefiks FROM tb_imbuhan");
    while($row=mysqli_fetch_array($result)){
        $cek_kata=mb_substr($stemed,0,1);
        if($cek_kata=== $row['prefiks']){
            $ketemu=true;
        }
        if($ketemu==true){
            $jml=mb_strlen($stemed);include_once'file';
            return$hasil=mb_substr($stemed,2,$jml);
        }else{return$hasil=$stemed;}
    }
}
```

Gbr 9. Pemotongan prefiks

Pada gambar 2 diatas merupakan fungsi *stem* prefiks yang membutuhkan 2 parameter.

2. Proses pemotongan infiks

```
function stemInfiks($koneksi,$stemed='') {
    $hasil=array();$in=array();
    $jml=mb_strlen($stemed);$nilai_awal=0;
    $result=mysqli_query($koneksi,"SELECT infiks FROM tb_imbuhan");
    while($row=mysqli_fetch_array($result)){
        for($i=0;$i<$jml;$i++){
            $potong=mb_substr($stemed,$nilai_awal,2);
            if(mb_substr($potong,0,1)==$row['infiks']){
            }else{
                if(!in_array($potong,$hasil)){
                    array_push($hasil,$potong);
                }
            }
            $nilai_awal=$nilai_awal+2;
        }
        $hasil_akhir=implode("", $hasil);
        return$hasil_akhir;
    }
}
```

Gbr 10. Pemotongan infiks

Pada gambar 3 diatas menunjukkan fungsi *stem* infiks yang membutuhkan 2 parameter yaitu variabel

koneksi ke *database* dan variabel *stemed*, inialisasi variabel hasil dengan nilai *array* kosong dan variabel *in* dengan *array* kosong juga lalu isi variabel *jml* dengan panjang dari variabel *stemed*, variabel nilai_awal dengan variabel 0 dan isi variabel *result* dengan nilai dari *database* pada *field* infiks.

3. Proses pemotongan sufiks

```
function stemSufiks($koneksi,$stemed='') {
    $ketemu=false;$hasil="";
    $jml=mb_strlen($stemed);
    $result=mysqli_query($koneksi,"SELECT sufiks FROM tb_imbuhan");
    while($row=mysqli_fetch_array($result)){
        $cek_kata=mb_substr($stemed,$jml-2,1);
        if($cek_kata=== $row['sufiks']){
            $ketemu=true;
        }
        if($ketemu==true){
            return$hasil=mb_substr($stemed,0,$jml-2);
        }else{return$stemed;}
    }
}
```

Gbr 11. Pemotongan sufiks

Pada gambar 4.26 diatas merupakan fungsi *stem* sufiks yang membutuhkan 2 parameter yaitu variabel koneksi ke *database* dan variabel *stemed* kemudian inialisasi variabel *ketemu* dengan *false* dan Variabel hasil dengan *string* kosong.

4. Proses pencarian brute force

```
function bruteForce($koneksi,$kataDasar='') {
    $hasil_akhir=null;
    // hapus baris kata dasar
    $kata_dasar1=array();
    $jml=mb_strlen($kataDasar);$n1=0;
    for($i=0;$i<$jml;$i++){
        $potong=mb_substr($kataDasar,$n1,1);
        array_push($kata_dasar1,$potong);
        $n1=$n1+2;
    }
    $kata_dasar=implode("", $kata_dasar1);
    $result1=mysqli_query($koneksi,"SELECT tsh_mashdar FROM tb_tashrif WHERE id_tsh = 14");
    $res=mysqli_fetch_array($result1);
    $alif=mb_substr($res['tsh_mashdar'],6,1);
    $kata_dasar.= $alif;
    // hapus baris kata mashdar
    $mashdar=array();
    $result=mysqli_query($koneksi,"SELECT id_tsh, tsh_mashdar FROM tb_tashrif");
    while($row=mysqli_fetch_array($result)){

```

```

$mashtdar_gabungan=array();
    $mashtdar2=array();$n1=0;

    $jml=mb_strlen($row['tsh_mashtdar']
);
    for($i=0;$i<$jml;$i++){

        $spotong=mb_substr($row['tsh_mashtda
r'],$n1,1);

array_push($mashtdar2,$spotong);
        $n1=$n1+2;
    }

    $mashtdar2_2=implode("",$mashtdar2);

    $mashtdar_gabungan['mashtdar']=$masht
dar2_2;

    $mashtdar_gabungan['key']=$row['id_
tsh'];

array_push($mashtdar,$mashtdar_gabun
gan);
}
$jml_mashtdar=count($mashtdar);
for($i=0;$i<$jml_mashtdar;$i++){

    if($kata_dasar==$mashtdar[$i]['masht
dar']){

        $shasil_akhir=$mashtdar[$i]['key'];
    }
}
    
```

Gbr 12. Pencarian brute force

D. Sampel Data Tashrif

No	FYU Mashtbi	FYU Mashtkari	Mashtdar	Isim FYU	Isim MaFuL	FYU Asma	FYU Nahi	Isim Makna	Isim Zaman	Isim Alar
1	قن	قن	قن	قن	قن	قن	قن	قن	قن	قن
2	قن	قن	قن	قن	قن	قن	قن	قن	قن	قن
3	قن	قن	قن	قن	قن	قن	قن	قن	قن	قن
4	قن	قن	قن	قن	قن	قن	قن	قن	قن	قن
5	قن	قن	قن	قن	قن	قن	قن	قن	قن	قن
6	قن	قن	قن	قن	قن	قن	قن	قن	قن	قن
7	قن	قن	قن	قن	قن	قن	قن	قن	قن	قن
8	قن	قن	قن	قن	قن	قن	قن	قن	قن	قن
9	قن	قن	قن	قن	قن	قن	قن	قن	قن	قن
10	قن	قن	قن	قن	قن	قن	قن	قن	قن	قن
11	قن	قن	قن	قن	قن	قن	قن	قن	قن	قن
12	قن	قن	قن	قن	قن	قن	قن	قن	قن	قن
13	قن	قن	قن	قن	قن	قن	قن	قن	قن	قن
14	قن	قن	قن	قن	قن	قن	قن	قن	قن	قن

Gbr 13. Sampel Data Tashrif

E. Implementasi Sistem

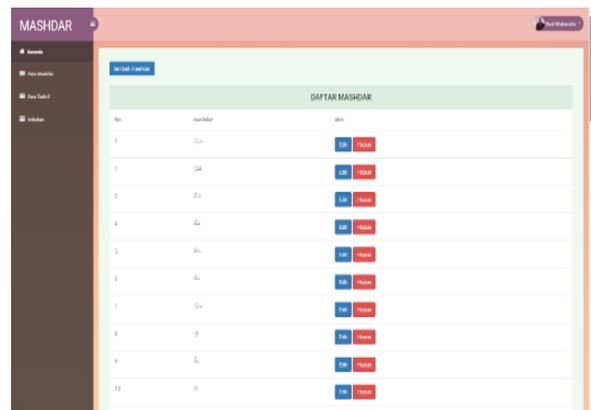
Adapun tampilan system kombinasi *brute force* dan *mashtdar* adalah sebagai berikut:



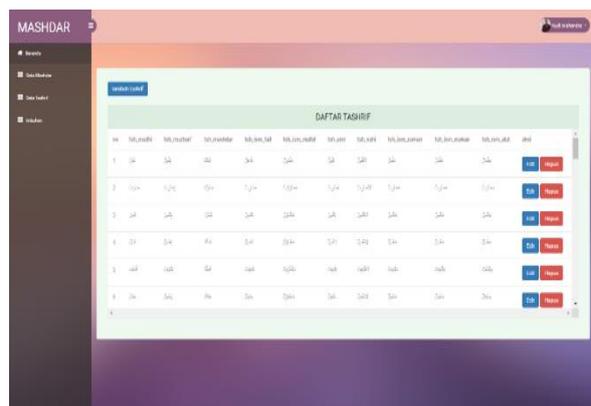
Gbr 14. Tampilan Menu Utama



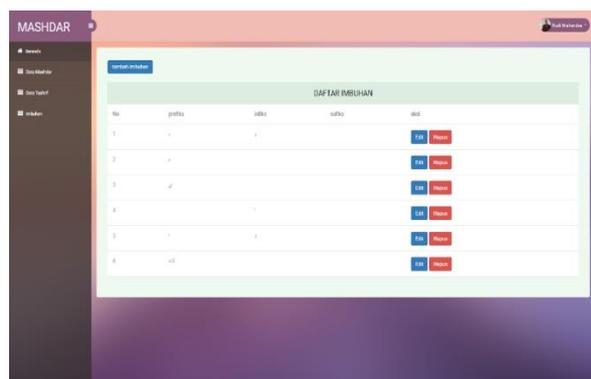
Gbr 15. Hasil pencarian



Gbr 16. Tambah Data Mashtdar



Gbr 17. Halaman Tashrif



Gbr 18. Halaman imbuhan

F. Pengujian Sistem

Hasil pengujian performa adalah:

a. Performa akurasi

$$\begin{aligned} \text{Akurasi} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \\ &= \frac{(11 + 4)}{(11 + 2 + 2 + 4)} \\ &= \frac{(11 + 4)}{19} = \frac{15}{19} \\ &= 0.789 \times 100\% \\ &= 78.9\% \end{aligned}$$

b. Performa *precision*

$$\begin{aligned} \text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}} \\ &= \frac{11}{(11 + 2)} = \frac{11}{13} \\ &= 0.846 \times 100\% = 84.6\% \end{aligned}$$

c. Performa *recall*

$$\begin{aligned} \text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}} \\ &= \frac{11}{(11 + 2)} = \frac{11}{13} \\ &= 0.846 \times 100\% = 84.6\% \end{aligned}$$

d. Performa *specificity*

$$\begin{aligned} \text{Specificity} &= \frac{\text{TN}}{\text{TN} + \text{FP}} \\ &= \frac{4}{(4 + 2)} = \frac{4}{6} \\ &= 0.666 \times 100\% = 66.6\% \end{aligned}$$

- [6] Pratiwi, H., Arfyanti, I., & Kurniawan, D. (2016). Implementasi Algoritma Brute Force Dalam Aplikasi Kamus Istilah Kesehatan. *jurnal ilmiah teknologi informasi terapan*, 119-125.
- [7] Mei M.L, A. R. (2015). *Studi Perbandingan Akurasi Light Stemming dan Khoja Stemming Pada Fi'il Madhi dan Mashdar*. Malang: Universitas Islam Negeri Maulana Malik Ibrahim.
- [8] Mesran. (2014). Implementasi Algoritma Brute Force Dalam Pencarian Data Katalog Buku Perpustakaan. *JurnalMajalah Ilmiah Informasi dan Teknologi Ilmiah (INTI)*, 100-104.

V. PENUTUP

Hasil dari kombinasi Algoritma *brute force* dan *stemming* diimplementasikan untuk mencari tingkat akurasi dalam pencarian kata *mashdar* beserta dengan perubahannya, dengan menggunakan data uji sebanyak 50 kata dalam teks Arab, sehingga akurasi yang diperoleh dari beberapa teks yang diambil sebagai *sample* pengujian mempunyai hasil akurasi 78.9%, *precision* 84.6%, *recall* 84.6% dan *specificity* 66.6%.

REFERENSI

- [1] Putri, E. K., & Setiadi, T. (2014). Penerapan Text Mining Pada Sistem Klasifikasi Email Spam Menggunakan Naive Bayes. *Jurnal Sarjana Teknik Informatika*, 73-83.
- [2] Pratita, H. S. (2016). Analisa Brute Force Attack menggunakan Scanning Aplikasi pada HTTP Attack. 2016.
- [3] Al Ajeeli, A. T. (2016). An Intelligent Framework for Natural Language Stems Processing. *Global Journal of Computer Science and Technology*, 16(1), 22-38.
- [4] Bin Ahmad Al-Ahdal, M. (2011). *Al Kawakib Ad-Durriyah*. Marocco: Haramain.
- [5] Herwijayanti, B., Ratnawati, D. E., & Muflikhah, L. (2018). Klasifikasi Berita Online Dengan Menggunakan Pembobotan TF-IDF dan Cosine Similarity. *Pengembangan Teknologi Informasi dan Komputer e-ISSN*, 2548, 964X.