

IMPLEMENTASI *SYLLABIFICATION ALGORITHM* DAN *STRING MATCHING ALGORITHM* UNTUK DETEKSI KESALAHAN PEMISAHAN SUKU KATA PADA E-PAPER ARTIKEL BERITA

Jati Sasongko Wibowo*, Mardi Siswo Utomo, Eko Nur Wahyudi

Fakultas Teknologi Informasi dan Industri, Universitas Stikubank Semarang

Email *jatisw@edu.unisbank.ac.id

ABSTRAK

Pemisahan suku kata pada dokumen digital sering kali dilakukan apabila saat menuliskan sebuah berita atau informasi sebuah kata harus di penggal menurut suku kata dikarenakan tempat yang tidak mencukupi. Apabila menggunakan tool atau aplikasi editor yang canggih biasanya pengaturan tulisan berkaitan dengan tempat dapat dilakukan secara otomatis tanpa harus memenggal sebuah kata menurut suku katanya. Dengan menggunakan tool yang bukan khusus untuk editor teks biasanya teks yang melebihi area tidak dapat secara otomatis menyesuaikan sehingga penulis berita akan memenggal kata untuk menyesuaikan tempat. Pemisahan suku kata yang dilakukan tidak sesuai dengan kaidah pemisahan suku kata dalam bahasa Indonesia, hanya berdasarkan masalah tempat yang penting kata tersebut masuk dalam area, sehingga terjadi pemisahan suku kata yang kurang pas di baca.

Data yang digunakan dalam penelitian ini menggunakan dokumen digital yang diambil dari sebuah website berita tribunnews. Dokumen digital dalam bentuk format image kemudian dirubah ke dalam format teks, selanjutnya diproses dimasukkan ke dalam basis data. Data yang sudah masuk dalam basis data di olah menggunakan metode *Syllabification* dan *String Matching*. Proses yang dihasilkan dapat mengetahui pemisahan yang sesuai dengan kaidah pemisahan bahasa Indonesia.

Jumlah kata yang mempunyai tanda hubung diperoleh dari data artikel sejumlah 2084, jumlah kata yang tanda hubungnya sudah sesuai silabifikasi sejumlah 1650, sedangkan jumlah kata yang tanda hubungnya belum sesuai silabifikasi sejumlah 434. Sehingga kesalahan dalam memisahkan suku kata yang belum sesuai silabifikasi sebanyak 20%.

Kata kunci: Silabifikasi, String Matching, Suku Kata, Tanda Hubung

1. PENDAHULUAN

1.1 Latar Belakang

Proses pemisahan suku kata berdasarkan suku kata dapat dilakukan dengan menggunakan algoritma silabifikasi. Telah banyak perkembangan algoritma silabifikasi, antara lain M. Hafiz, K. Rabiah, A, A. Azreen, dan A. M Taufik yang mengembangkan algoritma silabifikasi untuk bahasa Melayu dan H. Cuayáhuítl yang mengembangkan algoritma silabifikasi untuk bahasa Spanyol menggunakan pendekatan berdasarkan aturan. Aturan yang diimplementasikan ke dalam algoritme dihasilkan sesuai dengan aturan tata bahasa Spanyol. Hasil pengujian menunjukkan bahwa algoritma tersebut memiliki tingkat kesalahan sebesar 1,6%. R. Weerasinghe, A. Wasala, dan K. Gamage mengembangkan algoritme suku kata untuk Sinhala, bahasa resmi Sri Lanka. Algoritma yang dikembangkan juga menggunakan pendekatan aturan. Aturan-aturan ini dihasilkan berdasarkan aturan linguistik Sinhala. Hasil pengujian menunjukkan bahwa akurasi algoritma adalah 99,95%. Aturan-aturan ini dihasilkan berdasarkan morfologi dan struktur suku kata bahasa Melayu. Pengujian dilakukan dengan menggunakan tiga sumber pengumpulan data yang berbeda. Penelitian ini menghasilkan

akurasi 60,7% untuk pendataan BERNAMA, akurasi 77,4% untuk pendataan pendataan Kamus Dewan, dan akurasi 71,6% untuk pendataan tumpang tindih. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan algoritma silabifikasi kata dasar bahasa Indonesia. Algoritma yang dikembangkan menggunakan pendekatan aturan. Aturan ini diusulkan berdasarkan aturan segmentasi kata dalam Peraturan Menteri Pendidikan dan Kebudayaan Republik Indonesia Nomor 50 Tahun 2015 tentang Pedoman Umum Ejaan Bahasa Indonesia.

2. TINJUAN PUSTAKA

Tahapan penelitian yang dilakukan.

2.1. *Syllabification Algorithm*

- 1) Scan data berupa kata dari kiri ke kanan
- 2) Cek keberadaan huruf diftong ('ai', 'ei', 'au', dan 'oi') pada sebuah kata.
 - a. apabila huruf diftong berada di depan, maka berikan tanda indeks setelah huruf diftong
 - b. apabila huruf diftong berada di tengah, maka berikan tanda indeks sebelum dan setelah huruf diftong,
 - c. apabila huruf diftong berada di belakang, maka berikan tanda indeks setelah huruf diftong.
- 3) Pengecekan huruf vokal dan konsonan:
 - a. apabila urutan ke-i berupa huruf konsonan dan urutan ke-i+1 berupa huruf konsonan, berikan tanda indeks setelah huruf urutan ke-i.
 - b. apabila urutan ke-i berupa huruf vokal dan urutan ke-i+1 berupa huruf vokal, berikan tanda indeks setelah huruf urutan ke-i.
 - c. apabila urutan ke-i berupa huruf vokal dan urutan ke-i+1 berupa huruf konsonan dan urutan ke-i+2 berupa huruf vokal, berikan tanda indeks setelah huruf urutan ke-i
- 4) Lakukan pemisahan suku kata dengan memberikan tanda '-' sesuai urutan indeks yang telah ditentukan.

3. METODE PENELITIAN

3.1. *Crawling Data*

Mengumpulkan data berupa file dokumen digital dari sebuah situs berita dengan cara melakukan crawling data / mengunduh data pada website berita Tribunnews versi cetak pada tanggal 25 April 2018 dengan alamat website: <http://www.tribunnews.com/epaper>. Adapun file yang di unduh berupa file digital dalam format gambar dengan detail alamat url <https://s3-ap-southeast-1.amazonaws.com/tribun-3/epaper/jateng/1.jpg>, hingga <https://s3-ap-southeast-1.amazonaws.com/tribun-3/epaper/jateng/24.jpg>, mulai dari halaman 1 hingga halaman 24. Sehingga didapatkan file data dalam bentuk format jpg sejumlah 24 file.

3.2. *Konversi Jpg ke Pdf*

Melakukan konversi data dari file jpg menjadi file pdf menggunakan program nitro pro 9. Konversi dari file jpg menjadi file pdf harus dilakukan dikarenakan nitro pro 9 tidak dapat melakukan konversi file jpg secara langsung ke dalam format file txt, sehingga harus melalui file pdf terlebih dahulu. Sehingga pada tahapan ini di dapatkan 24 file dalam format pdf.

3.3. Konversi Pdf ke Txt

Melakukan konversi data dari file pdf menjadi file txt menggunakan program nitro pro 9, dengan hasil konversi ada beberapa kata yang belum terbaca dengan baik, dikarenakan ada beberapa sebab, diantaranya file pdf yang dihasilkan dari file jpg yang resolusinya atau dpinya masih terlalu rendah sehingga menghasilkan file pdf juga dengan kualitas text yang rendah pula, sebab lainnya ada beberapa text yang tertutup dengan warna atau gambar atau foto pada halaman beritanya sehingga text tidak dapat di recognition dengan baik. Kata yang belum terbaca dengan baik pada tahapan ini diabaikan, tidak dilakukan pembetulan, dikarenakan masih banyak teks yang masih terbaca dengan baik.



Gambar 1. Media Berita dalam Bentuk File Digital

2.5. Konversi Alternatif

Pada tahapan ini dilakukan alternatif konversi file jpg ke dalam format file txt dengan menggunakan aplikasi Microsoft one note 2016, sebagai pembanding dari hasil konversi dari program nitro pro 9. Hasil yang di dapat bahwa hasil konversi menggunakan aplikasi Microsoft one note 2016 lebih baik dari pada menggunakan program nitro pro 9. Indikator yang digunakan bahwa hasil konversi file jpg ke dalam format teks yang dihasilkan bahwa teks yang belum terbaca dengan sempurna jumlahnya lebih sedikit, atau dengan kata lain hasil konversinya lebih baik.

2.6. Punctuation Removal / Menghapus Tanda Baca

Punctuation Removal dilakukan untuk menghapus semua tanda baca pada dokumen txt, kecuali tanda hubung '-'. Dikarenakan tanda hubung ini yang akan dijadikan sebagai dasar pengujian untuk proses pendeteksian pemisahan suku kata yang benar sesuai kaidah silabifikasi. Sehingga hasil dari tahapan ini berupa data dokumen yang hanya terdiri dari huruf, spasi, dan tanda hubung, sedangkan tanda baca seperti titik, koma, titik koma, tanda tanya, tanda seru, dan lain sebagainya akan di hapus.

Perintah yang digunakan untuk proses punctuation removal, sebagai berikut:

```
$line = preg_replace("/[^a-zA-Z0-9\-\-]/+", "", $line);
```

2.7. Tokenisasi

Tahapan ini melakukan proses dari dokumen file dalam format txt di masukkan ke dalam database atau tabel dengan bentuk per kata, dengan cara pemisahannya berdasarkan spasi. Sehingga akan di dapatkan sejumlah data yang terdiri dari kata dan dikelompokkan berdasarkan asal dokumennya. Dalam basis data pengelompokkan dokumen dilakukan dengan cara memberikan kode dengan nama id, sedangkan data kata diberikan nama term. Maka setelah semua dokumen dimasukkan ke dalam basis data akan diketahui bahwa dokumen dengan id=1 berarti berasal dari dokumen 1, atau dari halaman 1 media berita dalam bentuk cetak. Sedangkan kata yang masuk dalam basis data akan diketahui berasal dari dokumen mana, dengan melihat id-nya. Data yang berhasil masuk ke dalam basis data dapat di rinci sebagai berikut:

Tabel 1. Dokumen dan Jumlah Kata

Dokumen	Jumlah Kata
1	643
2	1280
3	744
4	983
5	804
6	802
7	862
8	675
9	631
10	516
11	1276
12	730
13	629
14	709
15	865
16	752
17	975
18	1373
19	856
20	1256
21	976
22	296
23	564
24	278
Total	19475

Perintah yang digunakan untuk proses tokenisasi, sebagai berikut:

```
if ($start=1 && $line!="<text>") {
  $kata = preg_split("/[s]*/", $line);
  foreach($kata as $tok) {
    if (isset($freq[$tok])){
      $freq[$tok]+=1;
    }else{
      $freq[$tok]=1;}}}
```

2.8. Hasil Tokenisasi pada Aplikasi

Hasil tokenisasi yang ditampilkan pada aplikasi dapat di lihat sebagai berikut sekaligus source code yang digunakan untuk menampilkannya.

```
$query1 = "select count(*) from (select * from token1) as z";
$result1 = mysql_query($query1);
$query = "select * from token1";
$result = mysql_query($query);
$count = mysql_fetch_array($result1);
echo "jumlah record: $count[0]";
$numrows = mysql_num_rows($result);
```

Tabel 2 Hasil Tokenisasi

No	ID	Term	Freq
1	1	novanto	23
2	1	kpk	22
3	1	yang	17
4	1	dan	16
5	1	di	12
6	1	dengan	11
7	1	tahun	11
8	1	itu	10
9	1	pada	9
10	1	kasus	9
11	1	ke	9
12	1	korupsi	9
13	1	setya	8
14	1	hakim	8
15	1	vonis	8
16	1	dari	8
17	1	ketua	6
18	1	lebih	6
19	1	halaman	6
20	1	ini	6

Jumlah Record: 22659

2.9. Hapus Kata Tanpa Tanda Hubung

Hasil tokenisasi merupakan seluruh kata yang terdapat pada dokumen dimasukkan dalam database, termasuk kata yang mempunyai tanda hubung. Sehingga kata yang tanpa tanda hubung perlu di hapus. Hasil akhir dari tahapan ini berupa sekumpulan data berupa kata yang di dalamnya terdapat tanda hubung. Tampilan source code maupun hasil pada aplikasi dapat dilihat sebagai berikut.

Tabel 3 Hasil Penghapusan Kata Tanpa Tanda Baca

No	ID	Term	Freq
1	1	e-ktp	5
2	1	un-tuk	2
3	1	membawa-kan	2
4	1	per-caya	1
5	1	pikir-pikir	1
6	1	ketidak-sesuaian	1
7	1	didak-wakan	1
8	1	semi-final	1
9	1	ba-bak	1
10	1	champi-ons	1
11	1	pi-dana	1
12	1	de-ngan	1
13	1	ba-yern	1
14	1	terbayang-kan	1
15	1	per-nah	1
16	1	munc-hen	1
17	1	ku-asa	1
18	1	ha-kim	1
19	1	du-gaan	1
20	1	pengadil-an	1

Jumlah Record: 2084

```

$query1 = "select count(*) from (select * from token3) as z";
$result1 = mysql_query($query1);
$query = "select * from token3";
$result = mysql_query($query);
$count = mysql_fetch_array($result1);
echo "jumlah record: $count[0]";
$numrows = mysql_num_rows($result);

```

2.10. Kata tanpa Tanda Hubung

Kata yang sudah di proses dengan hasil berupa kata dengan tanda hubung, berikutnya di proses untuk dihilangkan tanda hubungnya. Sehingga hasil akhirnya berupa sekumpulan data kata yang tanpa tanda hubung. Berikut source code dan tampilan dalam aplikasinya.

Tabel 4 Hasil Penghapusan Tanda Hubung

No	ID	Term1	Term2	Freq
1	1	e-ktp	ektp	5
2	1	un-tuk	untuk	2
3	1	membawa-kan	membawakan	2
4	1	per-caya	percaya	1
5	1	pikir-pikir	pikirpikir	1
6	1	ketidak-sesuaian	ketidaksesuaian	1
7	1	didak-wakan	didakwakan	1
8	1	semi-final	semifinal	1
9	1	ba-bak	babak	1
10	1	champi-ons	champions	1
11	1	pi-dana	pidana	1
12	1	de-ngan	dengan	1
13	1	ba-yern	bayern	1
14	1	terbayang-kan	terbayangkan	1
15	1	per-nah	pernah	1

16	1	munc-hen	munchen	1
17	1	ku-asa	kuasa	1
18	1	ha-kim	hakim	1
19	1	du-gaan	dugaan	1
20	1	pengadil-an	pengadilan	1

Jumlah Record: 2084

2.11. Silabifikasi

Silabifikasi merupakan proses untuk memenggal sebuah kata berdasarkan suku kata, sesuai dengan kaidah bahasa Indonesia. Dalam tahap ini kata yang di penggal berdasarkan kaidah silabifikasi pada kolom term2. Pemisahan suku kata di tandai dengan penambahan tanda hubung. Sebagai contoh pada baris ke-11, pada kolom term2 terdapat kata pidana, yang apabila dilakukan proses silabifikasi menjadi pi-da-na.

Tabel 5 Hasil Silabifikasi

No	ID	Term1	Term2	Silabifikasi	Freq
1	1	e-ktp	ektp	ek-tp	5
2	1	un-tuk	untuk	un-tuk	2
3	1	membawa-kan	membawakan	mem-ba-wak-an	2
4	1	per-caya	percaya	per-ca-ya	1
5	1	pikir-pikir	pikirkikir	pi-kir-pi-kir	1
6	1	ketidak-sesuaian	ketidaksesuaian	ke-ti-dak-se-su-a-i-an	1
7	1	didak-wakan	didakwakan	di-da-kwak-an	1
8	1	semi-final	semifinal	se-mi-fi-nal	1
9	1	ba-bak	babak	ba-bak	1
10	1	champi-ons	champions	cham-pions	1
11	1	pi-dana	pidana	pi-da-na	1
12	1	de-ngan	dengan	de-ngan	1
13	1	ba-yern	bayern	ba-yern	1
14	1	terbayang-kan	terbayangkan	ter-ba-yangk-an	1
15	1	per-nah	pernah	per-nah	1
16	1	munc-hen	munchen	mun-chen	1
17	1	ku-asa	kuasa	ku-a-sa	1
18	1	ha-kim	hakim	ha-kim	1
19	1	du-gaan	dugaan	du-ga-an	1
20	1	pengadil-an	pengadilan	pe-nga-dil-an	1

Jumlah Record: 2084

```
$query1 = "select count(*) from (select * from token5) as z";
$result1 = mysql_query($query1);
$query = "select * from token5";
$result = mysql_query($query);
$count = mysql_fetch_array($result1);
echo "jumlah record: $count[0]";
$numrows = mysql_num_rows($result);
```

2.12. String Hyphens

String Hyphens menunjukkan posisi tanda hubung pada sebuah kata yang diambilkan dari data pada kolom term1. Posisi tanda hubung diikuti satu karakter di belakang dan satu karakter di depan tanda hubung. Sehingga hasil akhir akan terlihat tiga karakter dengan tanda hubung di tengah.

Tabel 6 Hasil Pengambilan String Tanda Hubung

No	ID	Term1	Term2	Silabifikasi	String_Hyphens	Freq
1	1	e-ktp	ektp	ek-tp	e-k	5
2	1	un-tuk	untuk	un-tuk	n-t	2
3	1	membawa-kan	membawakan	mem-ba-wak-an	a-k	2
4	1	per-caya	percaya	per-ca-ya	r-c	1
5	1	pikir-pikir	pikirkikir	pi-kir-pi-kir	r-p	1
6	1	ketidak-sesuaian	ketidaksesuaian	ke-ti-dak-se-su-a-i-an	k-s	1
7	1	didak-wakan	didakwakan	di-da-kwak-an	k-w	1
8	1	semi-final	semifinal	se-mi-fi-nal	i-f	1
9	1	ba-bak	babak	ba-bak	a-b	1
10	1	champi-ons	champions	cham-pions	i-o	1
11	1	pi-dana	pidana	pi-da-na	i-d	1
12	1	de-ngan	dengan	de-ngan	e-n	1
13	1	ba-yern	bayern	ba-yern	a-y	1
14	1	terbayang-kan	terbayangkan	ter-ba-yangk-an	g-k	1
15	1	per-nah	pernah	per-nah	r-n	1
16	1	munc-hen	munchen	mun-chen	c-h	1
17	1	ku-asa	kuasa	ku-a-sa	u-a	1
18	1	ha-kim	hakim	ha-kim	a-k	1
19	1	du-gaan	dugaan	du-ga-an	u-g	1
20	1	pengadil-an	pengadilan	pe-nga-dil-an	l-a	1

Jumlah Record: 2084

```

$query1 = "select count(*) from (select * from token5) as z";
$result1 = mysql_query($query1);
$query = "select * from token5";
$result = mysql_query($query);
$count = mysql_fetch_array($result1);
echo "jumlah record: $count[0]";
$numrows = mysql_num_rows($result);

```

2.13. String Hyphens Matching

String Hyphens Matching digunakan untuk memperlihatkan pemisahan suku kata yang tepat pada kata yang terdapat Term1 dibandingkan dengan kata pada kolom silabifikasi. Perbandingan yang di dapat antara kata pada kolom String_Hyphens_Matching dengan kata pada kolom Term Silabifikasi apabila terdapat data yang sama maka akan ditampilkan.

Tabel 7 Hasil Perbandingan yang Matching

No	ID	Term1	Term2	Silabifikasi	String_Matching	Freq
1	1	un-tuk	untuk	un-tuk	n-t	2
2	1	per-caya	percaya	per-ca-ya	r-c	1
3	1	pikir-pikir	pikirkikir	pi-kir-pi-kir	r-p	1
4	1	ketidak-sesuaian	ketidaksesuaian	ke-ti-dak-se-su-a-i-an	k-s	1
5	1	semi-final	semifinal	se-mi-fi-nal	i-f	1
6	1	ba-bak	babak	ba-bak	a-b	1
7	1	pi-dana	pidana	pi-da-na	i-d	1
8	1	de-ngan	dengan	de-ngan	e-n	1
9	1	ba-yern	bayern	ba-yern	a-y	1
10	1	per-nah	pernah	per-nah	r-n	1
11	1	ku-asa	kuasa	ku-a-sa	u-a	1
12	1	ha-kim	hakim	ha-kim	a-k	1

13	1	du-gaan	dugaan	du-ga-an	u-g	1
14	1	pengadil-an	pengadilan	pe-nga-dil-an	l-a	1
15	1	kem-belfx	kembelfx	kem-bel-fx	m-b	1
16	1	mere-ka	mereka	me-re-ka	e-k	1
17	1	zi-dane	zidane	zi-da-ne	i-d	1
18	1	menga-ku	mengaku	meng-a-ku	a-k	1
19	1	dipopu-lerkan	dipopulerkan	di-po-pu-lerk-an	u-l	1
20	1	rival-nya	rivalnya	ri-val-nya	l-n	1

Jumlah Record: 1650

```
$query1 = "select count(*) from (select id, term, term2, silabifikasi, concat(right(left(term,locate('-',term) - 1),1),
right(left(term,locate('-',term) + 1),2)) string_hyphens, frequency from token5 where silabifikasi like
concat('%',concat(right(left(term,locate('-',term) - 1),1), right(left(term,locate('-',term) + 1),2)),'%') as z";
$result1 = mysql_query($query1);
$query = "select id, term, term2, silabifikasi, concat(right(left(term,locate('-',term) - 1),1),
right(left(term,locate('-',term) + 1),2)) string_hyphens, frequency from token5 where silabifikasi like
concat('%',concat(right(left(term,locate('-',term) - 1),1), right(left(term,locate('-',term) + 1),2)),'%')";
$result = mysql_query($query);
$count = mysql_fetch_array($result1);
echo "jumlah record: $count[0]";
$numrows = mysql_num_rows($result);
```

2.14. String Hyphens Not Matching

String Hyphens Matching digunakan untuk memperlihatkan pemisahan suku kata yang tepat pada kata yang terdapat Term1 dibandingkan dengan kata pada kolom silabifikasi. Perbandingan yang di dapat antara kata pada kolom String_Hyphens_Matching dengan kata pada kolom Term Silabifikasi apabila terdapat data yang tidak sama maka akan ditampilkan.

Tabel 8 Hasil Perbandingan yang Tidak Matching

No	ID	Term1	Term2	Silabifikasi	String_Hyphens_Not_Matching	Freq
1	1	e-ktp	ektp	ek-tp	e-k	5
2	1	membawa-kan	membawakan	mem-ba-wak-an	a-k	2
3	1	didak-wakan	didakwakan	di-da-kwak-an	k-w	1
4	1	champi-ons	champions	cham-pions	i-o	1
5	1	terbayang-kan	terbayangkan	ter-ba-yangk-an	g-k	1
6	1	munc-hen	munchen	mun-chen	c-h	1
7	1	sedang-kan	sedangkan	se-dangk-an	g-k	1
8	1	ecoventi-on	ecovention	eco-ven-tion	i-o	1
9	1	pihak-pihak	pihakpihak	pi-ha-kpi-hak	k-p	1
10	1	melaku-kan	melakukan	me-la-kuk-an	u-k	1
11	1	mem-vonis	memvonis	me-mvo-nis	m-v	1
12	1	ktp-e	ktpe	ktpe	p-e	1
13	1	menunjuk-kan	menunjukkan	me-nun-jukk-an	k-k	1
14	1	biro-krasi	birokrasi	bi-rok-ra-si	o-k	1
15	2	rak-yat	rakyat	ra-kyat	k-y	2
16	2	pe-rempuan	perempuan	per-em-pu-an	e-r	2
17	2	undang-undang	undangundang	un-da-ngun-dang	g-u	1
18	2	diet-ah	dietah	di-e-tah	t-a	1
19	2	e-ktp	ektp	ek-tp	e-k	1
20	2	kehl-dan	kehdan	keh-ld-an	l-d	1

Jumlah Record: 434

```

$query1 = "select count(*) from (select id, term, term2, silabifikasi, concat(right(left(term,locate('-',term) - 1),1),
right(left(term,locate('-',term) + 1),2)) string_hyphens, frequency from token5 where silabifikasi not like
concat('%',concat(right(left(term,locate('-',term) - 1),1), right(left(term,locate('-',term) + 1),2)),'%') as z";
$result1 = mysql_query($query1);
$query = "select id, term, term2, silabifikasi, concat(right(left(term,locate('-',term) - 1),1),
right(left(term,locate('-',term) + 1),2)) string_hyphens, frequency from token5 where silabifikasi not like
concat('%',concat(right(left(term,locate('-',term) - 1),1), right(left(term,locate('-',term) + 1),2)),'%')";
$result = mysql_query($query);
$count = mysql_fetch_array($result1);
echo "jumlah record: $count[0]";
$numrows = mysql_num_rows($result);

```

4. KESIMPULAN

Kesimpulan yang dapat di ambil pada penelitian ini yaitu bahwa dengan menggunakan Syllabification Algorithm dan String Matching Algorithm dapat digunakan untuk mendeteksi pemisahan suku kata sesuai dengan kaidah bahasa Indonesia. Sehingga akan membantu penulis berita untuk dapat mengurangi atau menghilangkan kesalahan dalam pemisahan suku kata. Data yang digunakan dalam hal ini menggunakan dokumen digital yang diambil dari situs berita tribunnews dalam bentuk format file image yang kemudian dikonversi ke dalam bentuk format file teks. Hasil konversi masih ada teks yang belum terbaca dengan baik, yang dikarenakan resolusi file image yang cenderung kecil.

DAFTAR PUSTAKA

- [1] Setiawan, E. (no date) Kamus Besar Bahasa Indonesia (KBBI), Arti kata suku - Kamus Besar Bahasa Indonesia (KBBI) Online. Available at: <https://kbbi.web.id/suku> (Accessed: December 10, 2022).
- [2] Effendi, D., Hartono, T., Kurnaedi, A., 2009. Penerapan String Matching Menggunakan Algoritma Boyer-Moore Pada Translator Bahasa Pascal Ke C. *Majalah Ilmiah Unikom*. pp. 262 – 275.
- [3] Hafiz, M., Rabiah, K., A, Azreen, A., M Taufik, A., 2011. Syllabification algorithm based on syllable rules matching for Malay language. In: *Proceedings of The 10th WSEAS International Conference on Applied Computer and Applied Computational Science*, pp. 279–286.
- [4] Hernández-Figueroa, Z., Carreras-Riudavets, F.J., Rodríguez-Rodríguez, G., 2013. Automatic syllabification for Spanish using lemmatization and derivation to solve the prefix's prominence issue. *Expert Syst. Appl.* 40, 7122–7131. doi:10.1016/j.eswa.2013.06.056.
- [5] Marchand, Y., Adsett, C.R., Damper, R.I., 2009. Automatic syllabification in english: a comparison of different algorithms. *Lang. Speech* 52 (Pt 1), 1–27. doi:10.1177/0023830908099881.
- [6] Mayer, T., 2010. Toward a totally unsupervised, language-independent method for the syllabification of written texts. In: *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pp. 63–71.
- [7] Mulyawati, I., Subagio, R.T., Martha, D., 2017. Implementasi Metode String Matching Untuk Aplikasi Pengarsipan Dokumen (Studi Kasus: Smpn 3 Sumber Kab. Cirebon). *Jurnal Digit*. pp. 50–61.
- [8] Rogova, K., Demuynck, K., Van Compernelle, D., 2013. Automatic syllabification using segmental conditional random fields. *Comput. Ling. Neth.J.* 3, 34–48.