

KAMUS DIGITAL TANAMAN OBAT MENGGUNAKAN ALGORITME KNUTH MORRIS PRATT BERBASIS MOBILE

Arie Qur'ania¹⁾, Triastinurmiatiningsih²⁾, Erika Candra³⁾

^{1,2,3)} Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Pakuan, Jl. Pakuan PO Box 452, Bogor, 16143 Jawa Barat, Indonesia

Corresponding Author: gurania@unpak.ac.id

Abstrak

Kendala dalam penggunaan aplikasi kamus digital tanaman obat adalah sulit untuk mencari informasi yang pasti terhadap pencarian obat, karena kosakata dalam penggunaannya menghasilkan beberapa arti lain dan belum tentu sama dengan apa yang dicari. Penelitian ini bertujuan untuk mengembangkan penelitian sebelumnya tentang kamus digital tanaman obat menggunakan algoritme Rocchio berbasis mobile yang melakukan pencarian data tanaman obat dengan mengecek kedekatan query dalam database yang memiliki kelemahan pada kata-kata yang memiliki arti sama dengan kata berbeda, penggunaan teknik pencarian lain yaitu KMP diharapkan mendapatkan hasil yang optimal terhadap hasil pencarian berdasarkan kata kunci nama penyakit. Penelitian ini menggunakan teknik pencarian algoritme Knuth Morris Pratt (KMP) pada kamus digital tanaman obat dengan mencocokkan kata kunci sebagai inputan dengan string antara teks dan pola. Hasil pengujian dengan 30 kata kunci pada 640 data tanaman obat menghasilkan nilai presisi 92,4, akurasi dan recall sebesar 100%. Aplikasi kamus digital tanaman obat memuat kumpulan nama tanaman obat. Aplikasi ini dikembangkan untuk memudahkan pengguna dalam mengakses kegunaan tanaman obat berdasarkan kata kunci nama penyakit umum seperti asma, diare, demam, sakit kepala, diabetes, dan lain sebagainya.

Kata kunci: Kamus digital tanaman obat; teknik pencarian; khasiat tanaman obat; knuth morris pratt

Abstract

The obstacle in using the digital dictionary application of medicinal plants is that it is difficult to find exact information on drug searches, because the vocabulary in producing several other items is not necessarily the same as what you are looking for. This study aims to develop previous research on a digital dictionary of medicinal plants using a mobile-based Rocchio algorithm that searches medicinal plant data by querying proximity in a database that has weaknesses in words that have the same meaning with different words, using another search technique, namely KMP is expected to get optimal results on search results based on the keyword name of the disease. This study uses the Knuth Morris Pratt (KMP) algorithm search technique on a digital dictionary of medicinal plants by matching keywords as input with strings between text and patterns. The test results with 30 keywords on 640 medicinal plant data resulted in a precision value of 92.4, accuracy and recall of 100%. Digital dictionary of abundant medicinal plants. Applications collection of medicinal plant names. This application was developed to make it easier for users to access the use of medicinal plants based on keywords, names of common diseases such as asthma, diarrhea, fever, headache, diabetes, and so on.

Keywords: Digital dictionary of medicinal plants; search techniques; knuth morris pratt

1. Pendahuluan

Aplikasi kamus digital tanaman obat memuat kumpulan nama tanaman obat. Aplikasi ini dikembangkan untuk memudahkan pengguna dalam mengakses kegunaan tanaman obat berdasarkan kata kunci nama penyakit umum seperti asma, diare, demam, sakit kepala, diabetes, dan lain sebagainya. Kendala dalam penggunaan aplikasi kamus digital tanaman obat adalah sulit untuk mencari informasi yang pasti terhadap pencarian obat, karena kosakata dalam

penggunaannya menghasilkan beberapa arti lain dan belum tentu sama dengan apa yang dicari, misalnya pengguna ingin mencari tanaman obat untuk asma, tetapi tidak muncul karena kata yang ada dalam kamus data tidak ada kata asma tetapi adanya bengek, atau kata lain misalnya mencret yang identik dengan diare, dan kata-kata lain yang sebenarnya memiliki pengertian yang sama. Penelitian yang terkait dengan teknik pencarian pada khasiat tanaman obat sudah dilakukan peneliti pada tahun 2020 menggunakan metode Rocchio dengan menggunakan 200 data tanaman obat dan terbatas pada filter pencarian yang tidak terpusat pada kegunaan tanaman obat berdasarkan penyakit [1], sehingga pada penelitian ini peneliti menambahkan data sebanyak 640 tanaman obat dengan teknik pencarian yang lain yaitu algoritme KMP, dengan perbaikan pada proses query sehingga pencarian pada aplikasi kamus digital tanaman obat lebih terarah dengan penggunaan kosa kata yang tepat.

Tumbuhan dapat hidup dimana-mana termasuk di rumah, taman, dan lingkungan hutan. Dalam kehidupan masyarakat, pohon digunakan sebagai obat untuk menyembuhkan penyakit. Penggunaan jamu sebagai obat selalu menjadi kebutuhan masyarakat desa, Tanaman obat (herbal) dapat berperan dalam pertolongan pertama pada gangguan kesehatan dan terbukti ampuh menyembuhkan berbagai macam penyakit [2]. Hasil Riset Kesehatan Dasar (Riskesdas) tahun 2018 melaporkan 10 tahun ke atas yang mengkonsumsi obat tradisional sebanyak 38,7%, sejumlah 14,5% penderita hipertensi menggunakan tanaman obat. Tanaman obat yang ada di Indonesia terdiri atas 9.600 spesies berkhasiat untuk obat dan 300 spesies yang digunakan untuk pengobatan tradisional oleh industri obat tradisional. Tanaman obat tradisional digunakan masyarakat karena kemudahan mendapatkan tanaman dan mudah diramu sendiri [3].

Penelitian tentang aplikasi tanaman obat sudah dilakukan dengan membuat ensiklopedia tanaman obat, dimana pengguna memilih kategori tanaman obat yang sudah disediakan yang terdiri atas 3 (tiga) pilihan kategori, yaitu kategori buah, sayuran dan rempah-rempah [4]. Penelitian lain terkait tanaman obat dilakukan dengan menginventarisasi data tanaman obat di Bengkulu dengan mengenali ciri tanaman obat berdasarkan citra daun [5]. Penelitian selanjutnya tentang kamus digital tanaman obat menggunakan algoritme *Boyer Moore* yang bertujuan untuk membuat aplikasi kamus tanaman obat berbasis android yang bisa membantu mahasiswa ataupun masyarakat secara umum yang ingin mencari tanaman herbal secara praktis dan efisien. Pencarian dilakukan dengan algoritma *boyermoore* dengan mencocokkan karakter [6]. Ke tiga aplikasi tersebut belum disediakan teknik pencarian berdasarkan nama penyakit sehingga pengguna tidak bisa langsung memfilter tanaman obat yang berkhasiat untuk menyembuhkan penyakit sesuai dengan yang diinputkan pengguna.

Penelitian yang terkait dengan penggunaan algoritme KMP pada pencarian nama latin tumbuhan berdasarkan kata kunci (*keyword*) yang di-input-kan oleh pengguna sistem [7]. Penelitian tentang pencarian kata juga dilakukan dengan menggabungkan algoritme KMP dan Boyer Moore, hasil pencarian yang ditampilkan berupa file teks yang tersedia dalam Komputer dan informasi mengenai jumlah dari file teks yang tersedia dalam komputer [8]. Algoritme KMP juga digunakan untuk pencocokan kata pada aplikasi perbaikan kata yang salah pada dokumen karena kesalahan kata yang berupa singkatan yang tidak baku dan berupa kata yang tidak baku [9]. Algoritme KMP juga digunakan pada penelitian pencarian informasi rute bis Mayasari [10], pencarian susunan kata pada bahasa Inggris [11], pencarian data sinopsis film bioskop berdasarkan inputan kata kunci pada sistem [12]. Aplikasi penerjemahan bahasa Mandailing-Indonesia yang untuk mempermudah pengguna dalam proses translasi suatu kata dengan memanfaatkan algoritma KMP [13]. KMP untuk menganalisis bagaimana proses pencocokan string yang dihasilkan dan membandingkan sejauh mana nilai kemiripan dari beberapa judul yang sama dan serupa sehingga dapat memberikan suatu informasi yang efektif bagi mahasiswa [14], KMP untuk pencarian informasi yang ingin diketahui pengguna pada aplikasi budaya Betawi [15] dan KMP untuk pengembangan sistem pencarian pada tujuh kitab hadis menggunakan KMP [16].

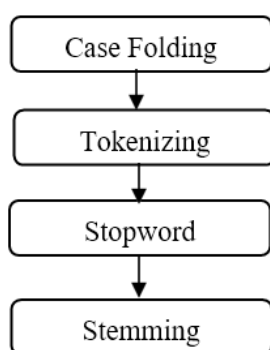
Penelitian ini mengembangkan penelitian sebelumnya tentang kamus digital tanaman obat menggunakan algoritma Rocchio berbasis mobile yang melakukan pencarian data tanaman obat dengan mengecek kedekatan query dalam database yang memiliki kelemahan pada kata-kata yang memiliki arti sama dengan kata berbeda, misalnya kata panas pada dasarnya hampir sama dengan demam pada atribut penyakit dan akan dicari tanaman obat yang berkhasiat untuk menjadi obat herbal [1]. Berdasarkan permasalahan pada input kata kunci dan output yang

dihasilkan, maka pada penelitian lanjutan ini akan menggunakan algoritma yang lain yaitu KMP untuk teknik pencarian pada kamus digital tanaman obat. Alasan penggunaan penerapan algoritma KMP karena dalam pencocokan string algoritma ini menyimpan informasi yang digunakan untuk melakukan jumlah pergeseran pencarian kata yang cocok, algoritma KMP menggunakan informasi tersebut untuk membuat pergeseran yang lebih jauh untuk mendapatkan informasi yang menyeluruh terhadap semua atribut dalam data tanaman obat. Sistem yang akan dibangun berbasis mobile android, inputan data tanaman yang akan diberikan oleh sistem kepada pengguna didapatkan melalui proses pencocokan string antara teks dan pola. Hasil dari penerapan algoritma KMP diharapkan dapat membantu permasalahan penggunaan kosakata yang diinputkan pengguna dalam aplikasi kamus digital tanaman obat.

2. Metode Penelitian

Data yang digunakan didapatkan dari buku kitab tanaman obat nusantara [17], serta data yang diambil dari sumber online yang terdiri dari photo tanaman, nama tanaman, nama_latin, nama_daerah, nama_lain, keluarga (*family*), zat_berkhasiat, penggunaan, pemerian, bagian yang digunakan, sediaan, waktu panen, penyimpanan dan data persamaan kemiripan penyakit. Proses pencarian berdasarkan nama penyakit atau penggunaannya dengan jumlah data yang digunakan sebanyak 640 data.

Penelitian ini menggunakan tahapan *preprocessing* yaitu tahapan penting dalam data *text mining*, memfokuskan pada data *cleaning*, termasuk menghilangkan *noise*. Tahapan *preprocessing* dilakukan untuk mendapatkan model yang dikehendaki. Tahapan *preprocessing* diantaranya *Case folding*, *Tokenizing*, *Stopword*, *Stemming*. Tahapan *preprocessing* ditunjukkan pada Gambar 1.



Gambar 1. Tahapan *preprocessing*

Penjelasan Gambar 1 sebagai tahapan *preprocessing* [18] adalah sebagai berikut :

1. *Case Folding* merupakan salah satu strategi umum yang dilakukan dalam proses teks, case folding yaitu mengubah semua huruf besar atau kapital menjadi huruf kecil. Contoh kalimat "Kencing nanah, kencing tidak lancar, bronchitis, produksi Air Susu Ibu (ASI)". Hasil pada tahapan case folding adalah "kencing nanah, kencing tidak lancar, bronchitis, produksi asi". Case folding tidak hanya mengubah huruf besar atau kapital menjadi huruf kecil tetapi juga menghilangkan tanda baca. Contoh kalimat "Penawar keracunan dipakai ±15 gram daun segar Allamanda cathartica". Hasil nya "penawar keracunan dipakai 15 15 gram daun segar allamanda cathartica".
2. *Tokenizing* adalah memotong kalimat menjadi beberapa bagian kata sekaligus membuang karakter tertentu, seperti tanda baca berdasarkan spasi (*input* : Peluruh haid dan obat keputihan *output* : Peluruh | haid | dan | obat | keputihan |). Token secara luas disebut sebagai istilah atau kata-kata pada saat proses tokenizing, secara bersamaan menghilangkan karakter tertentu seperti tanda baca, angka, dan karakter selain huruf alphabet, karena karakter-karakter tersebut dianggap sebagai pemisah kata (delimiter) dan tidak memiliki pengaruh terhadap proses teks. Beberapa kasus pemrosesan teks, angka tidak dihilangkan

karena masih dianggap penting. Token adalah turunan dari urutan karakter dalam beberapa dokumen tertentu yang dikelompokkan bersama sebagai unit semantik yang berguna selanjutnya untuk diproses.

3. Stopword atau filtering adalah tahap mengambil kata - kata penting dari hasil tokenizing. Proses filtering dapat menggunakan algoritma stoplist (membuang kata yang kurang penting) atau wordlist (menyimpan kata penting). Stoplist / stopwords adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan bag-of- words. Contoh stopwords adalah “yang”, “dan”, “di”, “dari” dan lain – lain
4. Stemming adalah proses pemetaan dan penguraian berbagai bentuk (variants) dari suatu kata menjadi bentuk kata dasarnya. Contoh : “pembengkakan”, hasil dari stemming “bengkak”.

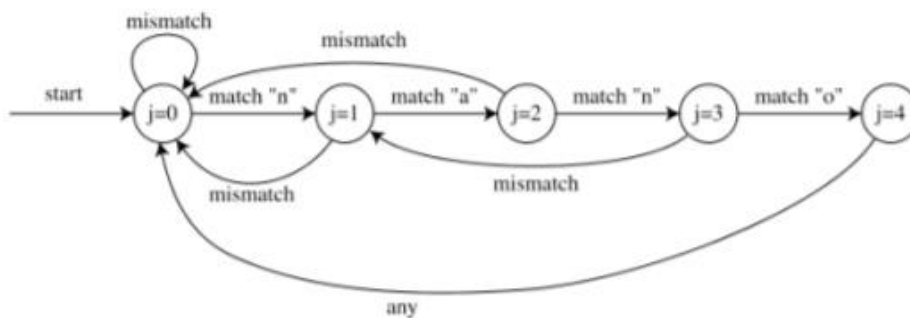
Algoritme KMP digunakan untuk proses pencarian nama penyakit pada kamus digital tanaman obat. Pseudocode KMP ditunjukkan pada Gambar 2.

```

j = 0;
for (i = 0; i < n; i++)
for (;;) { // loop until break
    if (T[i] == P[j]) { // matches?
        j++; // yes, move on to next state
        if (j == m) { // maybe that was the last state
            found a match;
            j = overlap[j];
        }
        break;
    } else if (j == 0) break; // no match in state j=0, give up
    else j = overlap[j]; // try shorter partial match
}
    
```

Gambar 2. Pseudocode KMP [19]

Langkah-langkah algoritme Knuth Morris Pratt dapat diilustrasikan seperti yang ditunjukkan pada Gambar 3 dengan proses simulasi dapat dilihat bahwa pencarian adalah dilakukan mulai dari string ke 0 hingga string ke n. Asalkan pencarian tetap dilakukan jika kondisi ketidakcocokan hingga string n.



Gambar 3. Finite automata algoritme KMP [20]

3. Hasil dan Pembahasan

3.1. Tahap Pre-KMP (Pre-Knuth-Morris-Pratt) merupakan suatu tahap pemberian nilai pergeseran terhadap keyword yang dimasukkan oleh user. Pemberian nilai ini dilakukan sebelum algoritma memulai pencocokkan string kedalam database, keyword yang dimasukkan akan dipecah menjadi array karakter. Tahap pre-KMP ini dapat mencegah pergeseran yang tidak berguna, proses pre-KMP hanya bergantung pada karakter yang

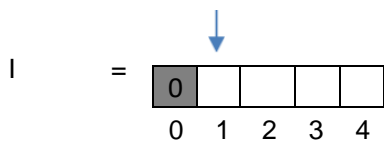
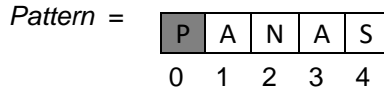
ada di dalam pattern, tidak bergantung kepada karakter di dalam teks. *Pseudocode* pre-KMP ditunjukkan pada Gambar 4.

```

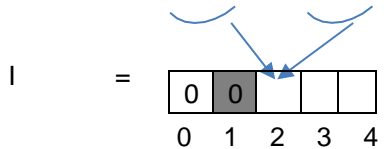
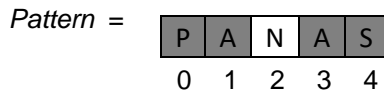
procedure preKMP(
  input P: array[0..n-1] of char,
  input n: integer,
  input/output kmpNext: array[0..n] of integer
)
Deklarasi:
i, j: integer
Algoritme
i := 0;
j := kmpNext[0] := -1;
while (i < n) {
  while (j > -1 and not(P[i] = P[j]))
    j := kmpNext[j];
  i := i+1;
  j := j+1;
  if (P[i] = P[j])
    kmpNext[i] := kmpNext[j];
  Else
    kmpNext[i] := j;
  Endif
Endwhile
    
```

Gambar 4. Pseudocode Pre- KMP [21]

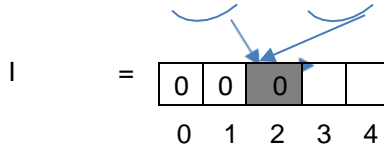
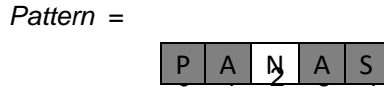
Contoh :



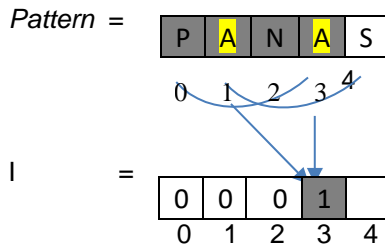
Langkah 1 : dimulai l [0], menjadi 0 karena tidak ada awalan atau akhiran yang tepat untuk satu string karakter.



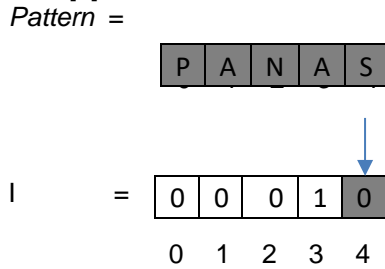
Langkah 2 : mengambil dua karakter dan memeriksa apakah ada substring yang merupakan awalan atau akhiran. Karena berbeda l [1] juga akan menjadi 0.



Langkah 3 : karena l [1] = 0, membandingkan karakter pertama dan terakhir dari string pattern nilai sebelumnya 0. berarti tidak ada karakter sama, jika l [2] maka nilai l [2] = 0.



Langkah 4 : mengambil 4 karakter, di sini karakter kedua $i[1]$ dan karakter $i[3]$ sama, jadi nilai $i[3] = 1$.



Langkah 5 : diakhir $i[4]$ karena tidak ada awalan atau akhiran yang tepat untuk satu string karakter $i[4]$ menjadi 0.

Tahap *pre-KMP* ini menghasilkan nilai *kmpNext* untuk masing-masing karakter pada *array keyword* yang dimasukkan oleh *user*. Nilai *kmpNext* tersebut selanjutnya akan digunakan sebagai nilai lompatan pergeseran pada tahap selanjutnya yaitu tahap *KMP-Search*.

3.2. Tahap KMP-Search

Tahap kedua setelah tahap *pre-KMP* pada algoritma Knuth-Morris-Pratt adalah tahap *KMP-Search*. Tahap ini merupakan proses pencocokkan string yang dimasukkan user yang telah dipecah menjadi *array* dan memiliki nilai pergeseran setiap karakternya, terhadap data yang terdapat pada database. Pseudocode *KMP search* dapat dilihat pada Gambar 5.

```

procedure KMPSearch(
    input m, n: integer,
    input P: array[0..n-1] of char,
    input T: array[0..m-1] of char,
    output ketemu: array[0..m-1] of boolean
)
Deklarasi:
i, j, next: integer
kmpNext: array[0..n] of integer

Algoritme:
preKMP(n, P, kmpNext)
i:=0
while (i<= m-n) do
    j:=0
    while (j < n and T[i+j] = P[j]) do
        j:=j+1
    Endwhile
    if(j >= n) then
        ketemu[i]:=true;
    Endif
    next:= j - kmpNext[j]
    i:= i+next
Endwhile
    
```

Gambar 5. Pseudocode KMP-Search

Berikut tahap KMP *search* :

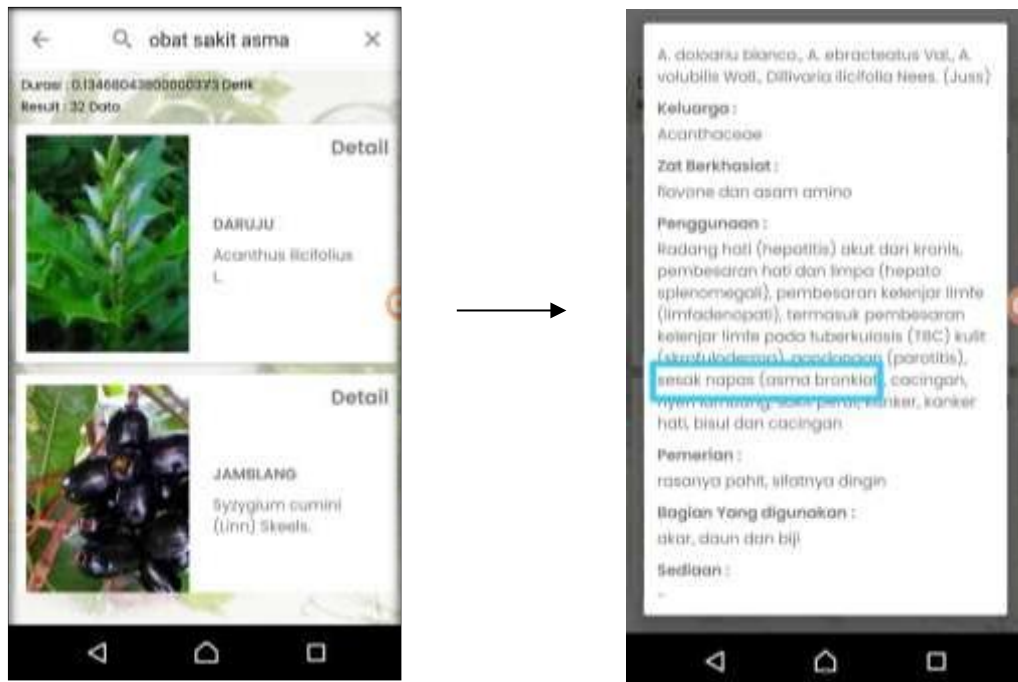
1. Mencari pola dalam string *text* akan menggunakan array *i* yang baru sajadibuat.
2. Menggunakan nilai dari *i* untuk memutuskan karakter berikutnya yang akan dicocokkan.

Aplikasi kamus digital tanaman obat memuat kumpulan nama tanaman obat. Seluruh data yang ada pada sistem disimpan dan dikelola menggunakan *MYSQLI*. Sistem yang dibangun berbasis *mobile* dengan pencarian berdasarkan nama penyakit. Inputan data tanaman yang akan diberikan oleh sistem kepada pengguna didapatkan melalui proses algoritme KMP yaitu pencocokan *string* antara teks dan pola dari kumpulan kata yang terdapat pada deksripsi penggunaan tanaman obat dengan kata yang sudah diproses *preprocessing*. Di dalam tahapan *Preprocessing* yaitu *tokenizing* dilakukan pemotongan kalimat menjadi beberapa kata, hasil dari *tokenizing* akan digunakan untuk pencocokan *KMP* dengan database persamaan. Berikut pembahasannya dapat dilihat pada Gambar 6 dan Gambar 7

Kasus 1

Inputan user : obat sakit asma
 Text : Radang hati (hepatitis) akut dan kronis, pembesaran hati dan limpa (hepato splenomegali), pembesaran kelenjar limfe (limfadenopati), termasuk pembesaran kelenjar limfe pada tuberkulosis (TBC) kulit (skrofuloderma), gondongan (parotitis), sesak napas (asma bronkial), cacingan, nyeri lambung, sakit perut, kanker, kanker hati, bisul dan cacingan
 Preprocessing : radang hati hepatitis akut kronis besar hati limpa hepato splenomegali besar kelenjar limfe limfadenopati besar kelenjar limfe tuberkulosis tbc kulit skrofuloderma gondong parotitis sesak napas asma bronkial cacing nyeri lambung perut kanker kanker hati bisul cacing.

Knuth morris pratt : asma
 Database persamaan : asma = ampek, bengek, manggah, sesak napas.



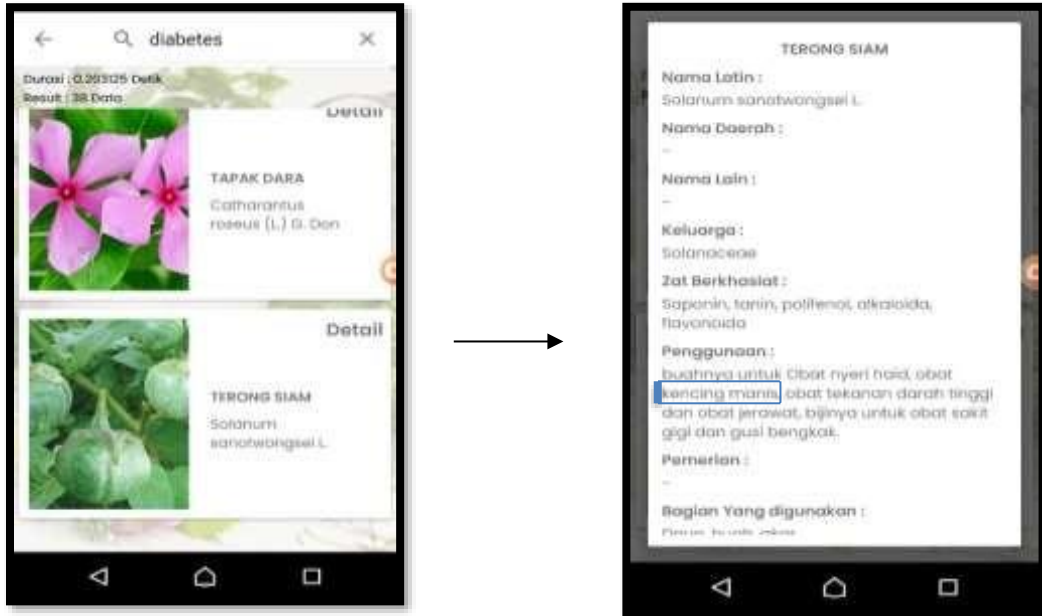
Gambar 6. Contoh pencarian kata penyakit asma

Kasus 2

keywords : diabetes

search : diabetes, kencing manis.

Result : Tanaman untuk penyakit atau penggunaan diabetes, kencing manis.



Gambar 7. Contoh pencarian kata diabetes

Contoh kasus untuk data yang tidak ditemukan

Data pada database terdiri dari photo tanaman, nama tanaman, nama_latin, nama_daerah, nama_lain, keluarga, zat_berkhasiat, penggunaan, bagian yang digunakan, sediaan, waktu panen, penyimpanan dan data persamaan kemiripan penyakit.

Knuth Moris pratt mencocokkan kata berdasarkan nama penyakit. Selain dari kata penyakit tidak dapat ditampilkan. Berikut contoh hasilnya yang ditunjukkan pada Gambar 8.



Gambar 8. Contoh kasus pada data yang tidak ditemukan

Penerapan algoritma knuth morris pratt pada sistem pencarian ini diperlukan suatu tolak ukur untuk mengukur kualitas hasil dokumen yang ditemukan kembali. Dari proses ini dapat dihasilkan 30 kata pencarian penggunaan yang berbeda – beda untuk dihitung nilai *precision* dan *recall* nya berdasarkan hasil pencarian. Hasil perhitungan ini menunjukkan evaluasi seberapa baik penerapan metode pada sistem dalam mencocokkan. Berikut hasil perhitungan *precision* dan *recall* dapat dilihat pada Tabel 1.

Tabel 1 Hasil perhitungan *precision* dan *recall*

No	K	A	B	C	D	P	R	AC
1	sakit perut	76	45	31	45	59	100	100
2	diare	79	79	0	79	100	100	100
3	peluruh haid	117	105	12	105	89	100	100
4	panas	150	148	2	148	98	100	100
5	asi	51	51	0	51	100	100	100
6	diabetes	38	38	0	38	100	100	100
7	penurun panas	157	139	18	139	88	100	100
8	asma	32	32	0	32	100	100	100
9	pusing	12	12	0	12	100	100	100
10	kencing manis	95	65	30	65	68	100	100
11	sakit kepala	37	31	6	31	83	100	100
12	kanker	16	16	0	16	100	100	100
13	jantung	10	10	0	10	100	100	100
14	haus	2	2	0	2	100	100	100
15	sariawan	46	46	0	46	100	100	100
16	pegal pegal	13	13	0	13	100	100	100
17	reumatik	22	22	0	22	100	100	100
18	Luka bakar	77	77	0	77	100	100	100
19	mata minus	31	31	0	31	100	100	100
20	cacing kremi	47	47	0	47	100	100	100
21	obat sakit gigi	44	25	19	25	56	100	100
22	obat jerawat	6	6	0	6	100	100	100
23	Obat kurap	4	4	0	4	100	100	100
24	menambah nafsu makan	47	47	0	47	100	100	100
25	masuk angin	12	10	2	10	83	100	100
26	panu	5	5	0	5	100	100	100
27	mimisan	12	12	0	12	100	100	100
28	cacingan	47	47	0	47	100	100	100
29	memar	18	18	0	18	100	100	100
30	meredakan jantungberdebar	20	10	10	10	50	100	100
	Mean					92.4	100	100

Keterangan

K : Keyword

A : Hasil keseluruhan dokumen yang di dapatkan pada aplikasi

- B : Hit (Hasil relevan pada aplikasi)
- C : Miss (Hasil yang tidak relevan pada aplikasi)
- D : Hit Database (Hasil yang relevan pada database)
- P : Precision dalam %
- R : Recall dalam %
- AC : Acuraccy %

Hasil dari pengujian yang didapat dari 30 data yang diujikan dengan rata-rata penggunaan metode KMP precision 92,4%, recall 100 %, dan acuraccy 100%. Berdasarkan hasil evaluasi kinerja sistem bahwa algoritme KMP dalam pencarian pencocokan kata adalah baik.

4. Kesimpulan

Penelitian ini menerapkan algoritme KMP pada pencarian kamus digital tanaman obat yaitu algoritme pencocokan string yang merupakan algoritma yang digunakan untuk melakukan pencarian sebuah *string* yang terdiri dari beberapa karakter yang biasa disebut *pattern*. Data tanaman berisikan informasi tanaman khasiat obat, nama latin tanaman obat, nama daerah tanaman obat, nama lain tanaman obat, keluarga tanaman obat, zat berkhasiat, penggunaan, pemerian, bagian yang digunakan, sediaan, waktu panen, dan penyimpanan tanaman obat dan data persamaan penyakit disimpan dan dikelola di database MYSQLI. *Preprocessing* adalah digunakan untuk pemrosesan isi dokumen sebelum dibandingkan. Tujuannya untuk membersihkan data dari noise yang terdiri dari imbuhan, angka, simbol dan karakter yang tidak relevan serta kata-kata yang tidak penting. Di dalam tahapan *preprocessing* yaitu *tokenizing* dilakukan pemotongan kalimat menjadi beberapa kata, hasil dari *tokenizing* akan digunakan untuk pencocokan KMP dengan database persamaan.

Sistem yang akan dibangun berbasis mobile android, Inputan data tanaman yang diberikan oleh sistem kepada pengguna didapatkan melalui proses pencocokan *string* antara teks dan pola dari pengecekan karakter yang dimasukkan oleh user dengan karakter yang ada pada database sehingga pencarian banyak menghasilkan *match*. Algoritme KMP memiliki dua tahap pada proses pencariannya, yaitu tahap pre-KMP yang merupakan tahap pemberian nilai pergeseran pada masing- masing karakter yang dimasukkan (*pattern*) oleh *user*. Tahap kedua yaitu KMP *Search* yang merupakan tahap pencocokkan karakter yang dimasukkan (*pattern*) oleh user dengan karakter yang ada pada database (*text*) berdasarkan nilai pergeseran. KMP menyimpan informasi yang digunakan untuk melakukan jumlah pergeseran, algoritma knuth morris pratt menggunakan informasi tersebut untuk membuat pergeseran yang lebih jauh. Knuth morris pratt diuji kinerja sistem dengan mengetahui nilai *accuracy* 100% nilai *recall* 100 % dan *preccision* dengan rata rata yang didapat *preccision* 92.4%. Berdasarkan hasil evaluasi kinerja sistem bahwa algoritme KMP dalam pencarian pencocokan menghasilkan nilai yang baik.

Referensi

- [1] A. Qur'ania, Triastinurmiatiningasih, and N. M. Ikhbal, "Kamus Digital Tanaman Obat menggunakan Algoritma Rocchio berbasis Mobile," *KOMPUTASI*, vol. 17, no. 2, pp. 354–361, 2020.
- [2] D. Harefa, "Pemanfaatan Hasil Tanaman Sebagai Tanaman Obat Keluarga (TOGA)," *Madani Indones. J. Civ. Soc.*, vol. 2, no. 2, pp. 28–36, 2020, doi: 10.35970/madani.v2i2.233.
- [3] H. Idris, "Back To Nature (Memanfaatkan Tanaman Obat Keluarga (TOGA))." p. 37, 2019.
- [4] D. E. Sari, S. Puspasari, and H. Sunardi, "Rekayasa Aplikasi Ensiklopedia Tanaman Obat Berbasis Android," *J. Ilm. Inform. Glob.*, vol. 9, no. 1, pp. 32–39, 2018, doi: 10.36982/jig.v9i1.461.
- [5] D. Maulidiya and K. Kasrina, "Pengembangan Modul Inventarisasi Tanaman Obat Pada Sistem Informasi Tanaman Obat Bengkulu," *J. Media Infotama*, vol. 12, no. 2, 2017, doi: 10.37676/jmi.v12i2.411.

- [6] Sumarni, Sutardi, and A. R. Saputra, "Aplikasi Kamus Tanaman Obat Menggunakan Algoritma Boyer Moore Berbasis Android," *SemanTIK*, vol. 3, no. 2, pp. 245–250, 2017.
- [7] A. Anofrizen, "Perancangan Sistem Pencarian Nama Latin Tumbuhan Berbasis," *J. Sains, Teknol. dan Ind.*, vol. 17, no. 2, pp. 59–64, 2020.
- [8] M. M. Y. Daeli and R. K. Hondro, "Perancangan Aplikasi Pencarian Kata dengan Kombinasi Algoritma Knuth Morris Pratt dan Algoritma Boyer Moore," *Maj. Ilm. INTI*, vol. XII, no. 2, pp. 271–275, 2017, [Online]. Available: <https://ejurnal.stmik-budidarma.ac.id/index.php/inti/article/view/380/362>.
- [9] S. Suwandana, "Pengembangan Aplikasi Perbaikan Kata Pada Dokumen Dengan Menerapkan Metode Knuth Morris Pratt," *Jursima*, vol. 4, no. 2, p. 47, 2016, doi: 10.47024/js.v4i2.141.
- [10] F. X. Syahasta A.T. and S. Hansun, "Rancang Bangun Aplikasi Informasi Rute Bus Mayasari Bakti Dengan Algoritma Knuth- Morris-Pratt Berbasis Android," *J. Inform.*, vol. 12, no. 1, 2016, doi: 10.21460/inf.2016.121.443.
- [11] S. H. Sampurno, "Aplikasi Permainan Susun Kata Untuk Pembelajaran Bahasa Inggris Dengan Algoritma Knuth–Morris–Pratt Dan Fisher-Yates Shuffle," *Matics*, vol. 7, no. 2, p. 83, 2016, doi: 10.18860/mat.v7i2.3283.
- [12] N. Novianti, R. C. G. I. Kembaren, D. M. Br Bangun, and N. Marbun, "Implementasi Algoritma Knuth Morris Pratt Pada Aplikasi Sinopsis Film Bioskop Berbasis Web," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 3, no. 1, pp. 398–401, 2019, doi: 10.30865/komik.v3i1.1619.
- [13] R. K. Hondro, Z. A. Hsb, and R. D. Sianturi, "Aplikasi Penerjemahan Bahasa Mandailing-Indonesia," *JURIKOM (Jurnal Ris. Komputer)*, vol. 3, no. 4, pp. 49–53, 2016.
- [14] W. Astuti, "Analisis String Matching Pada Judul Skripsi Dengan Algoritma Knuth-Morris Pratt (Kmp)," *Ilk. J. Ilm.*, vol. 9, no. 2, pp. 167–172, 2017, doi: 10.33096/ilkom.v9i2.136.167-172.
- [15] M. A. Firmanto, "Aplikasi Budaya Betawi Berbasis Android Menggunakan Algoritma Knuth Morris Pratt (Kite Betawi)," *J. Ilmu Tek. dan Komput.*, vol. 3, no. 2, p. 105, 2019, doi: 10.22441/jitkom.2020.v3.i2.004.
- [16] D. K. Sari, D. Sakethi, and R. Prabowo, "Pengembangan Sistem Pencarian Pada Tujuh Kitab Hadis Menggunakan Algoritma Knuth-Morris-Pratt," *J. Komputasi*, vol. 7, no. 1, 2019, doi: 10.23960/komputasi.v7i1.1703.
- [17] Herlina Widyaningrum dan Tim Solusi Alternatif , "Kitab tanaman obat Nusantara", Yogyakarta : Media Pressindo, 2019
- [18] I. Y. R. Pratiwi, R. A. Asmara, and F. Rahutoo, "Study of hoax news detection using naïve bayes classifier in Indonesian language," *Proc. 11th Int. Conf. Inf. Commun. Technol. Syst. ICTS 2017*, vol. 2018-January, no. February 2018, pp. 73–78, 2018, doi: 10.1109/ICTS.2017.8265649.
- [19] Sedgewick, R., & Wayne, K. 2015. Algorithms, (Deluxe): Book and 24-Part Lecture Series. Addison-Wesley Professional
- [20] Levitin, A., & Mukherjee, S. 2003. Introduction to the design & analysis of algorithms (p. 576). Reading: Addison-Wesley.
- [21] G. H. Ekaputri and Y. A. Sinaga, "Aplikasi Algoritma Pencarian String Knuth-Morris-Pratt dalam Permainan Word Search," *Tek. Inform. Inst. Teknol. Bandung*, pp. 2–4, 2006.