

KAMUS DIGITAL TANAMAN OBAT MENGGUNAKAN ALGORITMA ROCCHIO BERBASIS MOBILE

Arie Qur'ania¹⁾, Triastinurmiatiningsih²⁾, Nazar Muhamad Ikhbal³⁾
^{1,3)}Program Studi Ilmu Komputer, ²⁾Program Studi Biologi, FMIPA,
Universitas Pakuan, Bogor, Indonesia

Corresponding Author: qurania@unpak.ac.id

Abstrak

Kamus digital sudah banyak digunakan untuk memudahkan dalam pengolahan kata dan pencarian kata melalui media digital seperti handphone. Masyarakat umumnya mengetahui khasiat dan cara meracik tanaman obat dari pengalaman orang tua terdahulu atau melalui buku dan tulisan. Penelusuran melalui buku atau tulisan memerlukan waktu yang tidak singkat dibandingkan dengan penelusuran melalui media digital salah satunya adalah kamus digital. Penelitian bertujuan untuk membuat kamus digital tanaman obat berbasis mobile yang memiliki fasilitas pencarian berdasarkan kata yang diinputkan misalnya kandungan yang dimiliki oleh tanaman obat. Aplikasi kamus digital tanaman obat menggunakan metode pencarian dengan algoritma Rocchio dengan teknik relevance feedback untuk mengecek kedekatan query ke rata-rata dokumen relevan dengan tingkat perhitungan similaritas melalui tahapan tokenizing, filtering, stemming, dan Term Weighting dengan jumlah data sebanyak 200 tanaman obat.

Kata kunci: *kamus_digital, tanaman_obat, algoritma_Rocchio*

Abstract

Digital dictionaries have been widely used to facilitate word processing and word search through digital media such as mobile phones. Society generally knows the efficacy and how to mix medicinal plants from the experience of previous parents or through books and writings. Searching through books or writings requires a short time compared to searching through digital media, one of which is a digital dictionary. The research aims to create a digital dictionary of mobile-based medicinal plants which has a search facility based on the words entered, for example, the contents of the medicinal plants. The digital dictionary application of medicinal plants uses a search method with the Rocchio algorithm with relevance feedback techniques to check the proximity of the query to the average document relevant to the level of similarity calculation through the stages of tokenizing, filtering, stemming, and Term Weighting with a total data of 200 medicinal plants.

Keywords: *digital_dictionary, plant_feeds, algorithm_Rocchio*

1. Pendahuluan

Tanaman Obat di Indonesia digunakan dan dimanfaatkan secara luas sebagai pengobatan herbal, bahan jamu-jamuan, makan penguat daya tahan tubuh, produk kecantikan (kosmetik dan spa serta sebagai bahan baku industri makanan dan minuman. Di Indonesia, volume perdagangan obat tradisional tahun 2002 baru mencapai 150 juta dolar. Lebih dari 61% penduduk Indonesia sudah terbiasa mengkonsumsi obat tradisional, misalnya jamu. Sementara itu, kebutuhan bahan baku untuk perusahaan obat tradisional baik industri besar maupun kecil sebagian besar (85%) berasal dari hasil hutan dan pekarangan tanpa upaya budi daya (IPB Magazine, 2015)[1]. Penelitian mengenai tanaman obat sudah dilakukan oleh peneliti bersama tim pada tahun 2014 tentang identifikasi tanaman obat menggunakan citra daun dengan teknik kode fraktal [2], yang merupakan pengembangan penelitian sebelumnya yang menggunakan dimensi fraktal pada tahun 2012 [3]. Penelitian tersebut belum sampai pada khasiat dan penggunaan tanaman obat. Selama ini untuk pencarian informasi tentang khasiat tanaman obat

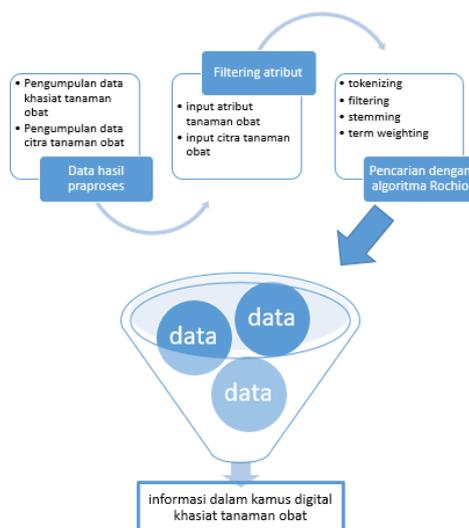
masih menggunakan manual, yaitu dengan pencarian menggunakan buku atau literatur, hal ini memerlukan waktu pencarian yang tidak singkat. Di era digital dengan kemajuan peralatan komputer, diperlukan sebuah aplikasi berbasis mobile yang dapat membantu masyarakat dalam mencari informasi hanya dengan proses yang cepat dan akurasi yang tinggi sesuai dengan apa yang dicari pengguna.

Penelitian ini bertujuan untuk membuat kamus digital tentang khasiat tanaman obat berbasis mobile yang memiliki fasilitas pencarian berdasarkan kata kunci penyakit atau nama tanaman obat. Di era mobile yang semakin pesat, informasi terkait tentang khasiat tanaman obat dikembangkan dalam bentuk aplikasi yang memudahkan masyarakat umum mencari manfaat serta cara penggunaan tanaman obat untuk mengobati penyakit dengan bahan alami. Aplikasi kamus khasiat tanaman obat dibangun dengan bantuan komputer yang diprogram menggunakan algoritma pencarian *Rocchio* yang akan mencari berdasarkan data yang sudah diinputkan ke dalam sistem database. Output berupa detail nama tanaman dan khasiat serta cara penggunaan dan bagian mana dari tanaman yang dapat dimanfaatkan untuk mengobati penyakit. Algoritma *Rocchio* digunakan dalam teknik pencarian data didasarkan pada kemampuan umpan balik relevansi menggunakan *Vector Space Model* yang didasarkan pada asumsi bahwa pengguna dalam pencarian data memiliki konsep umum tentang dokumen mana yang bisa dinyatakan relevan atau tidak relevan [5].

Penelitian yang berkaitan dengan pengembangan metode *Rocchio* diantaranya Najib, 2018 [4] menghasilkan Similaritas Dokumen Tugas Akhir Menggunakan Metode *Rocchio* sebuah metode yang dapat digunakan untuk membandingkan dokumen terhadap kesamaan isi antara data dengan merepresentasikan semua data dalam sebuah vector suatu *term* (istilah kata). *Rocchio* memiliki konsep umum pada dokumen yang relevan dan non-relevan sebagai sarana meningkatkan pencarian untuk mengetahui kemiripan dokumen dengan dokumen lainnya. Kristanda, 2017 [5] menghasilkan Rancang Bangun Aplikasi UMN Library Catalog Menggunakan Metode *Rocchio Relevance Feedback* menunjukkan bahwa pencarian buku di SLiMS UMN termasuk mudah untuk dilakukan dan mudah untuk dipelajari. SLiMS yang merupakan sistem informasi berbasis *web* ini juga berpotensi untuk didukung dengan aplikasi *mobile* dalam melakukan pencarian buku. Hal ini diperkuat dengan hasil survey terhadap ketertarikan responden dalam melakukan pencarian buku lewat *smartphone*.

2. Metode Penelitian

Tahap awal penelitian adalah dengan pengumpulan data tentang tanaman obat sebanyak 200 data tanaman obat. Data gambar dan khasiat tanaman obat yang sudah diinput akan difilter atribut sesuai dengan kata kunci yang biasa diinputkan untuk pencarian khasiat tanaman obat. Proses pencocokan kata kunci dan data tanaman obat akan diproses dalam tahap tokenizing, filtering, stemming, dan term weighting. Aplikasi kamus digital akan dipersiapkan bersamaan dengan proses input data ke database. Algoritma pencarian dengan *Rocchio* akan disusun untuk bisa membandingkan antara kata kunci yang diinput pada aplikasi dengan dokumen yang memiliki kemiripan tertinggi. Proses selanjutnya adalah proses filter data tanaman obat yang berelasi dengan kata kunci yang diinput. Hasil filter data tanaman obat akan dimunculkan sebagai output dari hasil pencarian. Tahapan penelitian ditunjukkan pada Gambar 1.



Gambar 1. Tahapan Penelitian

2.1. Tanaman obat

Tanaman obat menurut Hariana, 2008 [6] merupakan tanaman yang memiliki khasiat obat untuk pencegahan dan menyembuhkan penyakit. Tanaman obat yang tumbuh dan dibudidayakan di Indonesia dan digunakan secara turun temurun untuk tujuan kesehatan disebut herbal asli Indonesia, menurut peraturan menteri kesehatan RI no 6 tahun 2016 [7]. Tanaman obat atau biofarmaka didefinisikan sebagai jenis tanaman yang sebagian, seluruh tanaman dan atau eksudat tanaman tersebut digunakan sebagai obat, bahan atau ramuan obat-obatan. Eksudat tanaman adalah isi sel yang secara spontan keluar dari tanaman atau dengan cara tertentu sengaja dikeluarkan dari selnya. Eksudat tanaman dapat berupa zat-zat atau bahan-bahan nabati lainnya yang dengan cara tertentu dipisahkan/diisolasi tanamannya menurut Herdiani, 2012 [8]. Tanaman obat potensial adalah tanaman yang diduga mengandung atau memiliki senyawa aktif berkhasiat obat tetapi belum dibuktikan penggunaannya secara ilmiah-medis sebagai bahan obat-obatan. Contoh buah mengkudu dan temu kunci [9].

2.2. Kamus Digital

Kamus digital adalah kamus yang lebih mengutamakan pada fasilitas pengolah kata elektronik, yaitu sebuah fasilitas yang memungkinkan aplikasi pengolah kata memeriksa ejaan dari dokumen yang diketik. Pengguna kamus elektronik atau kamus digital dalam aplikasi pemrosesan teks merupakan hal yang tidak dapat dihindarkan. Kamus merupakan basis pemeriksaan, basis pengetahuan, bahkan sebagai basis penyelidikan [10].

2.3. Algoritma Rocchio

Pencarian dengan algoritma Rocchio merupakan teknik Metode *rocchio relevance feedback* adalah strategi reformulasi *query* yang digunakan untuk membantu *user* pemula dalam *information retrieval systems*. *User* disajikan dengan hasil pencarian dokumen yang relevan [11]. *Information retrieval* atau *document retrieval* adalah teknik untuk menghasilkan hasil dokumen relevan dengan *request* dari user melalui perbandingan *request* dengan indeks yang menghasilkan tingkat *similarity* dari konten yang dipilih [12]. Teknik *relevance feedback* digunakan setelah *user* memasukkan *keyword* yang data sudah didapat dari pencarian di database. *Relevance feedback* mendekati *query* ke rataan dokumen relevan dengan tingkat perhitungan similaritas [13]. Berikut tahapan perhitungan metode Rocchio.

- a. *Tokenizing* adalah tahap pemrosesan sebuah kata kunci menjadi unit kecil. Pembuatan token dilakukan pada kata kunci dan dokumen yang didapat.
- b. *Filtering* adalah tahap mengambil kata-kata penting dari hasil *tokenizing*. Tahap ini kata-kata yang tidak deskriptif dibuang, seperti kata “yang”, “dan”, “di”.

- c. *Stemming* adalah tahap mencari akar kata sesuai dengan kata kunci. Tahap ini dilakukan proses pengembalian berbagai kata ke dalam suatu representasi.
- d. *Term Weighting* adalah tahap pembobotan tiap term yang dicari pada setiap dokumen sehingga dapat diketahui ketersediaan dan kemiripan suatu term dalam dokumen. Perhitungan *term frequency* dan *inverse document frequency* akan dilakukan bersamaan dengan *query* yang merupakan fitur di dalam *database*. *Term frequency* (tf) adalah frekuensi dari kemunculan sebuah *term* dalam dokumen yang bersangkutan. *Inverse document frequency* (idf) merupakan sebuah perhitungan dari bagaimana *term* didistribusikan secara luas pada koleksi dokumen yang bersangkutan. Rumusan nilai idf sebuah term dapat dijabarkan menjadi persamaan berikut [14].

$$idf = \log \frac{n}{df} \tag{1}$$

keterangan:

idf : nilai *inverse document frequency*.

n : jumlah dokumen di dalam koleksi.

df : nilai *document frequency*.

Jika pencarian pada dokumen yang cukup besar, skema yang dapat digunakan untuk pemberian bobot adalah *term frequency* dikalikan dengan *inverse document frequency* yang disebut sebagai nilai bobot *term* atau β [13].

$$\beta = (tf) * idf \tag{2}$$

Keterangan:

β : nilai bobot *term*.

tf: nilai *term frequency*.

idf: nilai *inverse document frequency*.

Selanjutnya, tingkat kemiripan term dengan judul dokumen akan dimasukkan ke dalam sebuah fungsi perhitungan similaritas yaitu metode Rocchio yang dapat dilihat dari rumusan manual sebagai berikut [13].

$$R = N + \beta \left(\left(\frac{Dp}{Np} \right) - \left(\frac{Dn}{Nn} \right) \right) \tag{3}$$

Keterangan:

R : tingkat kemiripan term.

N : jumlah term tiap dokumen.

β : nilai bobot term.

Dp : term dari dokumen relevan.

Np : jumlah term keseluruhan dokumen relevan.

Dn : term dari dokumen tidak relevan.

Nn : jumlah term keseluruhan dokumen tidak relevan

3. Hasil dan Pembahasan

Algoritma *Rocchio* melibatkan suatu query yang diproses menjadi sebuah *vektor space* yang dapat dibandingkan dengan masing-masing centroid kelas yang ada pada korpus. Pendekatan dilakukan dengan mencari kemiripan dua *vektor space*, vektor query dianggap mirip dengan sebuah centroid kelas menggunakan jarak (*distance*) atau menggunakan kemiripan (*similarity*). Jika menggunakan jarak, yang dicari adalah kelas yang memiliki jarak yang terkecil dengan query. Tahapan penerapan algoritma *Rocchio* seperti contoh berikut :

3.1. Contoh Penerapan Algoritma *Rocchio*

Proses awal dengan pencarian kata pada Query : Glukosa pada 3 dokumen, proses *tokenizing*, *filtering*, dan *stemming* ditunjukkan pada Tabel 1.

Tabel 1. Tahapan *tokenizing*, *filtering*, dan *stemming* pada dokumen

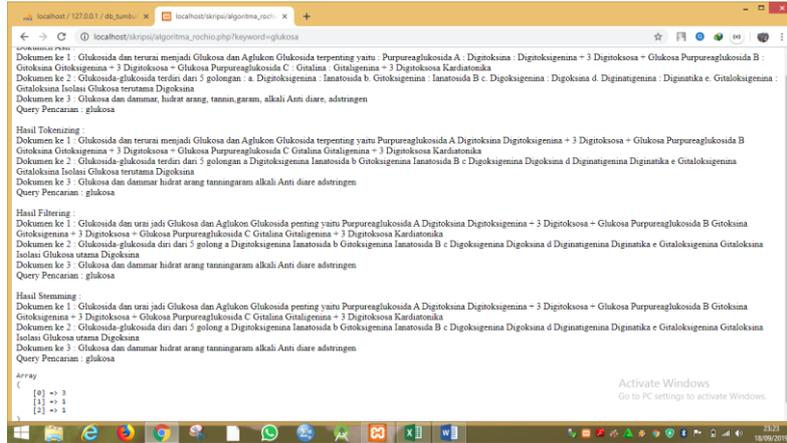
| No | Isi Dokumen | <i>Tokenizing</i> | <i>Filtering</i> | <i>Stemming</i> |
|----|---|---|---|---|
| 1 | Glukosida dan terurai menjadi Glukosa dan Aglukon Glukosida | Glukosida dan terurai menjadi Glukosa dan Aglukon Glukosida | Glukosida dan urai jadi Glukosa dan Aglukon Glukosida penting yaitu | Glukosida dan urai jadi Glukosa dan Aglukon Glukosida penting yaitu |



| | | | | |
|---|--|--|--|--|
| | terpenting yaitu : Purpureaglukosida A : Digitoksina : Digitoksigenina + 3 Digitoksosa + Glukosa Purpureaglukosida B : Gitoksina Gitoksigenina + 3 Digitoksosa + Glukosa Purpureaglukosida C : Gitalina : Gitaligenina + 3 Digitoksosa Kardiatonika | terpenting yaitu Purpureaglukosida A Digitoksina Digitoksigenina + 3 Digitoksosa + Glukosa Purpureaglukosida B Gitoksina Gitoksigenina + 3 Digitoksosa + Glukosa Purpureaglukosida C Gitalina Gitaligenina + 3 Digitoksosa Kardiatonika | Purpureaglukosida A Digitoksina Digitoksigenina + 3 Digitoksosa + Glukosa Purpureaglukosida B Gitoksina Gitoksigenina + 3 Digitoksosa + Glukosa Purpureaglukosida C Gitalina Gitaligenina + 3 Digitoksosa Kardiatonika | Purpureaglukosida A Digitoksina Digitoksigenina + 3 Digitoksosa + Glukosa Purpureaglukosida B Gitoksina Gitoksigenina + 3 Digitoksosa + Glukosa Purpureaglukosida C Gitalina Gitaligenina + 3 Digitoksosa Kardiatonika |
| 2 | Glukosida- glukosida terdiri dari 5 golongan : a. Digitoksigenina : lanatosida b. Gitoksigenina : lanatosida B c. Digoksigenina : Digoksina d. Diginatigenina : Diginatika e. Gitaloksigenina : Gitaloksina Isolasi Glukosa terutama Digoksina | Glukosida- glukosida terdiri dari 5 golongan a Digitoksigenina lanatosida b Gitoksigenina lanatosida B c Digoksigenina Digoksina d Diginatigenina Diginatika e Gitaloksigenina Gitaloksina Isolasi Glukosa terutama Digoksina | Glukosida- glukosida diri dari 5 golong a Digitoksigenina lanatosida b Gitoksigenina lanatosida B c Digoksigenina Digoksina d Diginatigenina Diginatika e Gitaloksigenina Gitaloksina Isolasi Glukosa utama Digoksina | Glukosida- glukosida diri dari 5 golong a Digitoksigenina lanatosida b Gitoksigenina lanatosida B c Digoksigenina Digoksina d Diginatigenina Diginatika e Gitaloksigenina Gitaloksina Isolasi Glukosa utama Digoksina |
| 3 | Glukosa dan dammar, hidrat arang, tannin,garam, alkali Anti diare, adstringen | Glukosa dan dammar hidrat arang tanningaram alkali Anti diare adstringen | Glukosa dan dammar hidrat arang tanningaram alkali Anti diare adstringen | Glukosa dan dammar hidrat arang tanningaram alkali Anti diare adstringen |

Tabel 1 menunjukkan pada proses *tokenizing* menghilangkan tanda baca seperti tanda koma, titik dua, dan titik, pada proses *filtering* menghilangkan awalan dan akhiran seperti pada kata terurai menjadi kata urai, sedangkan pada proses *stemming* proses pengembalian berbagai kata ke dalam suatu representasi, pada kasus ini hasil pada proses *filtering* sama dengan hasil *stemming*.

Hasil proses Rocchio selengkapnya ditunjukkan pada Gambar 2 dengan contoh Query yang diinput 'Glukosa' maka akan terkumpul dokumen-dokumen yang mengandung kata 'Glukosa'.



Gambar 2. Penerapan algoritma Rocchio

Proses perhitungan untuk mendapatkan nilai Tf-Idf ditunjukkan pada Tabel 2, sedangkan hasil perhitungan jarak ditunjukkan pada Tabel 3.

Tabel 2. Contoh hasil perhitungan Tf-Idf

| Term | TF | | | df | D/df | Log (D/df) | w | | | |
|-------------------|----|----|----|----|------|------------|-------|-------|-------|-------|
| | Q | D1 | D2 | | | | D3 | Q | D1 | D2 |
| Glukosida | 0 | 1 | 2 | 0 | 3 | 1 | 0 | 0 | 0 | 0 |
| Glukosa | 1 | 3 | 1 | 1 | 5 | 0,6 | 0,221 | 0,221 | 0,221 | 0,221 |
| Aglukon | 0 | 1 | 0 | 0 | 1 | 3 | 0,477 | 0 | 0,477 | 0 |
| Purpureaglukosida | 0 | 3 | 0 | 0 | 3 | 1 | 0 | 0 | 0 | 0 |
| Digitoksina | 0 | 1 | 0 | 0 | 1 | 3 | 0,477 | 0 | 0,477 | 0 |
| Digitoksigenina | 0 | 1 | 1 | 0 | 2 | 1,5 | 0,176 | 0 | 0,176 | 0,176 |
| Diigitoksosa | 0 | 3 | 0 | 0 | 3 | 1 | 0 | 0 | 0 | 0 |
| Gitalina | 0 | 1 | 0 | 0 | 1 | 3 | 0,477 | 0 | 0,477 | 0 |
| Gitaligenina | 0 | 1 | 0 | 0 | 1 | 3 | 0,477 | 0 | 0,477 | 0 |
| Kardiatonika | 0 | 1 | 0 | 0 | 1 | 3 | 0,477 | 0 | 0,477 | 0 |
| Ilanatosida | 0 | 1 | 1 | 0 | 2 | 1,5 | 0,176 | 0 | 0,176 | 0,176 |
| Digoksina | 0 | 0 | 1 | 0 | 1 | 3 | 0,477 | 0 | 0 | 0,477 |
| Isolasi | 0 | 0 | 1 | 0 | 1 | 3 | 0,477 | 0 | 0 | 0,477 |
| Danmar | 0 | 0 | 0 | 1 | 1 | 3 | 0,477 | 0 | 0 | 0 |
| Hidrat | 0 | 0 | 0 | 1 | 1 | 3 | 0,477 | 0 | 0 | 0 |
| Arang | 0 | 0 | 0 | 1 | 1 | 3 | 0,477 | 0 | 0 | 0 |
| Alkari | 0 | 0 | 0 | 1 | 1 | 3 | 0,477 | 0 | 0 | 0 |

Tabel 2 menunjukkan jumlah term pada dokumen 1, 2, dan 3 (D1, D2, D3), misalnya kata 'Glukosa' pada D1 ditemukan sebanyak 3, pada D2 sebanyak 1, dan pada D3 sebanyak 1. Proses selanjutnya menghitung nilai bobot (w) yang akan menghitung kedekatan dengan nilai bobot pada Query (Q), dalam kasus ini nilai w pada term 'Glukosa' menghasilkan nilai yang sama yaitu 0,221.

Hasil nilai w pada Tabel 2 digunakan untuk menghitung panjang vektor pada term yang dicari, pada kasus ini term "Glukosa" memiliki jarak terdekat sebesar 0,05 jika dibandingkan dengan nilai pada term yang lain. Hasil perhitungan seperti ditunjukkan pada Tabel 3.

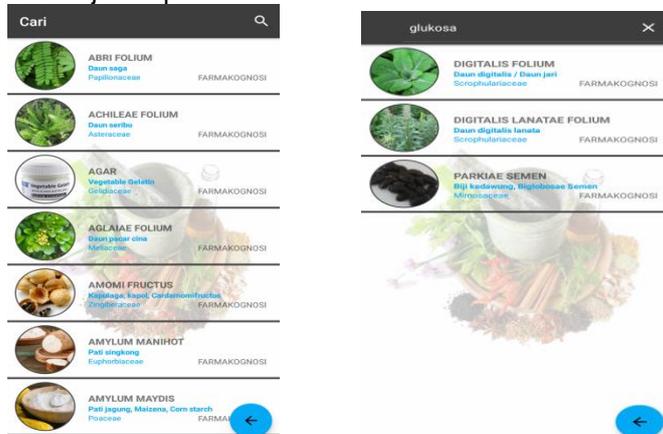
Tabel 3. Menghitung jarak dan Query

| Term | W(Q)*W(Di) | | | Panjang Vektor | | | |
|-------------------|------------|------|------|----------------|-------|------|------|
| | D1 | D2 | D3 | Q^2 | D1^2 | D2^2 | D3^2 |
| Glukosida | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Glukosa | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 |
| Aglukon | 0 | 0 | 0 | 0 | 0.227 | 0 | 0 |
| Purpureaglukosida | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Digitoksina | 0 | 0 | 0 | 0 | 0.277 | 0 | 0 |

| | | | | | | | |
|-----------------|-------------|-------------|-------------|-------------|--------------|--------------|--------------|
| Digitoksigenina | 0 | 0 | 0 | 0 | 0.031 | 0.031 | 0 |
| Diigitoksosa | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Gitalina | 0 | 0 | 0 | 0 | 0.277 | 0 | 0 |
| Gitaligenina | 0 | 0 | 0 | 0 | 0.277 | 0 | 0 |
| Kardiatonika | 0 | 0 | 0 | 0 | 0.277 | 0 | 0 |
| Ianatosida | 0 | 0 | 0 | 0 | 0.031 | 0.031 | 0 |
| Digoksina | 0 | 0 | 0 | 0 | 0 | 0.277 | 0 |
| Isolasi | 0 | 0 | 0 | 0 | 0 | 0.277 | 0 |
| Dammar | 0 | 0 | 0 | 0 | 0 | 0 | 0.277 |
| Hidrat | 0 | 0 | 0 | 0 | 0 | 0 | 0.277 |
| Arang | 0 | 0 | 0 | 0 | 0 | 0 | 0.277 |
| Alkali | 0 | 0 | 0 | 0 | 0 | 0 | 0.277 |
| SUM | 0.05 | 0.05 | 0.05 | 0.05 | 1.497 | 0.666 | 1.158 |

3.2. Pencarian Data

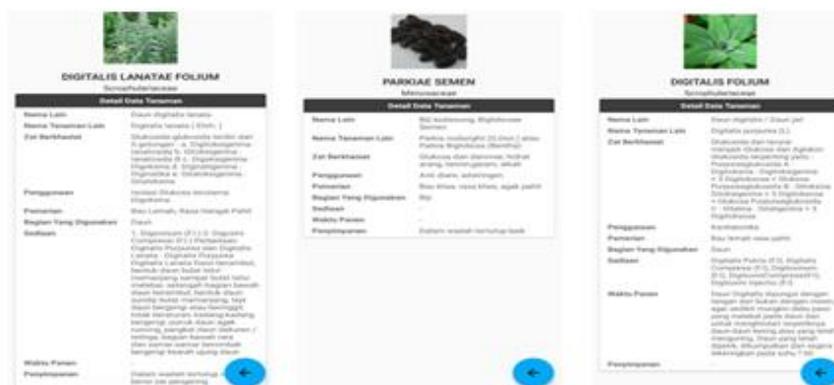
Pencarian data pada aplikasi dengan cara menginputkan kata yang akan dicari, misalnya akan dicari kata glukosa, maka akan muncul beberapa data tumbuhan yang mempunyai unsur kata 'Glukosa' seperti ditunjukkan pada Gambar 3.



Gambar 3. Contoh pencarian kata 'Glukosa' pada kamus digital

Hasil yang ditunjukkan pada Gambar 3 adalah proses memfilter data tanaman obat yang mengandung unsur kata yang dicari, dalam contoh ini kata 'Glukosa' yang diinputkan dalam kamus digital menampilkan sebanyak tiga data tanaman obat yaitu tanaman obat Digitalis Folium, Digitalis Lanatae Folium, dan Parkiae Semen. Khasiat dan detail dari tanaman obat bisa ditampilkan dalam kamus digital seperti ditunjukkan pada Gambar 4. Hasil akurasi pencarian kata menggunakan algoritma Rochio mencapai 100 persen berdasarkan pengujian pada setiap kata yang diinput, kata yang diinput bisa muncul di semua field selama kata yang dicari memang ada dalam database.





Gambar 4 Detail Data Tanaman

4. Kesimpulan

Kesimpulan dari penerapan algoritma Rocchio untuk pencarian data tanaman obat dapat memudahkan pengguna dalam mencari detail tentang nama daerah tanaman, khasiat, cara meracik, serta bagian tanaman yang dapat digunakan untuk mengobati penyakit tanpa harus membaca satu per satu data seperti data yang tertulis dalam buku atau kamus manual tanaman obat. Aplikasi dibuat menggunakan android yang memungkinkan pengguna bisa menginstall di dalam perangkat mobile yang bisa digunakan secara realtime dengan menginputkan kata kunci sesuai dengan kebutuhan pengguna.

References

- [1] IPB Magazine. 2015. http://ipbmag.ipb.ac.id/orasiilmiah/c310cca1b6c979754e8d90b8f187b8d9/Guru-Besar-IPB-Tahun_2050-Nilai-Tanaman-Obat-Mencapai-Lima-Triliun-Dolar. IPB Press. Bogor
- [2] Harsani P, Qurania A, Triastinuriatiningsih. 2014. Pengembangan Web Services Identifikasi Tanaman menggunakan Kode Fraktal dalam Sistem Informasi Tanaman Obat Indonesia. Proceeding Seminar Nasional Teknologi Informasi Komunikas dan Manajemen. Palembang.
- [3] Harsani P, Mulyana, M, Prasetyorini. 2012. Application of Image Retrieval Using Fractal Dimension ti Identify Medicinal Plant. Proceeding Internasional Seminar on Science Technology Innovations 2012. ISBN 1978-602-95064-5-7.
- [4] Najib (2018), Similaritas Dokumen Tugas Akhir Menggunakan Metode Rocchio.
- [5] Kristanda (2017). Rancang Bangun Aplikasi UMN Library Catalog Menggunakan Metode Rocchio Relevance Feedback.
- [6] Hariana, A., 2008, Tumbuhan Obat dan Khasiatnya, Penebar Swadaya, Jakarta.
- [7] Peraturan Menteri Kesehatan Republik Indonesia Nomor 6 tahun 2016 tentang FormulariumObat Herbal Asli Indonesia.
- [8] Herdiani, E. (2012). Potensi Tanaman Obat Indonesia. Online. Tersedia: <http://www.bbpp-lembang.info/index.php/arsip/artikel-pertanian/585-potensitanaman-obat-indonesia>.
- [9] Hidayat, S dan Team Flora. 2008. "Khasiat Herbal". Gramedia Jakarta
- [10] Harfatiani, Rina Rizky. 2007. Teknik Riset Operasi. Surabaya: Kartika. Hal. 37.
- [11] Selberg, E. W., "Information Retrieval Advances Using Relevances Feedback". *Department of Computer Science and Engineering University of Washington*, 2011.
- [12] Liddy, E. D., "Automatic Document Retrieval. Encyclopedia of Language and Linguistic. 2nd Ed". 2011. Philadelphia.
- [13] Yugianus, P., Dachlan, H. S., dan Hasanah, R. N., Pengembangan Sistem Penelusuran Katalog Perpustakaan dengan Metode Rocchio Relevance Feedback. *Jurnal Electrics Communications Controls Informatics Systems (EECCIS)*. 2013. Vol. 7 No. 1.
- [14] Uden dan Mark, V., "Rocchio Relevance Feedback in Learning Classification Algorithms". *Department of Computing Science University of Nijmegen*. 2011.