Pemanfaatan Metode *Vector Space Model* dan Metode *Cosine Similarity* pada Fitur Deteksi Hama dan Penyakit Tanaman Padi

Ana Triana
Informatika, Fakultas MIPA, Universitas Sebelas Maret Surakarta
Jl. Ir. Sutami No 36 A Surakarta
anaatriana@student.uns.ac.id

Ristu Saptono
Informatika, Fakultas MIPA, Universitas Sebelas Maret Surakarta
Jl. Ir. Sutami No 36 A Surakarta
r_saptono@uns.ac.id

Meiyanto Eko Sulistyo
Informatika, Fakultas MIPA, Universitas Sebelas Maret Surakarta
Jl. Ir. Sutami No 36 A Surakarta
mekosulistyo@uns.ac.id

ABSTRAK

Hama dan penyakit yang menyerang tanaman padi dapat diperkirakan dari beberapa gejala yang sering dialami. Namun untuk memastikan hal tersebut, perlu dilakukan deteksi hama dan penyakit tanaman padi dengan membandingkan gejala yang dialami dengan gejala-gejala yang ada dalam fitur pendeteksi ini. Untuk membandingkannya, dibutuhkan feedback atau input jawaban dari pengguna.

Banyaknya gejala dari semua daftar hama dan penyakit membuat pengguna harus memberikan feedback sebanyak daftar semua gejala yang ada, maka pada proses masukan gejala pada fitur ini dibuat dengan berbasis tekstual sehingga pengguna dapat langsung memberikan feedback tanpa harus menjawab satu per satu apakah gejala-gejala itu dialami atau tidak. Untuk mendapatkan daftar gejala yang sesuai dengan feedback, maka digunakan metode Vector Space Model dengan menentukan kemiripan diantara keduanya. Hasil output dari metode Vector Space Model inilah yang nantinya akan digunakan selanjutnya menggunakan metode Cosine Similarity untuk mendeteksi hama dan penyakit tanaman padi yang sesuai dengan feedback tersebut.

Dari 25 percobaan yang telah dilakukan untuk pengujian metode Vector Space Model dalam mengidentifikasi input feedback menghasilkan akurasi sebesar 96% dan akurasi sebesar 100% setelah dilakukan pengujian terhadap metode Cosine Similarity dalam mendeteksi hama dan penyakit tanaman padi.

Kata kunci

Cosine Similarity, Deteksi Hama dan Penyakit, Feedback, Gejala, Vector Space Model

1. PENDAHULUAN

Salah satu permasalahan serius yang dihadapi akibat dari penanaman tanaman padi secara monokultur adalah meningkatnya siklus hama dan penyakit tanaman padi. Seiring dengan banyaknya akses mencari informasi mengenai pertanian, petani dapat dengan mudah memperkirakan masalah apa yang sedang terjadi pada tanaman padinya. Namun kesalahan mengambil keputusan atau kurang terpercayanya informasi yang didapat bisa berakibat tidak tepatnya cara penanganan masalah yang sedang terjadi, sehingga petani memerlukan suatu perangkat yang dapat digunakan untuk mendeteksi hama dan penyakit berdasarkan gejala-gejala yang dialami.

Pendeteksian penyakit dapat dilakukan oleh petani dengan cara mencocokkan gejala yang dialami dengan gejala penyakit yang terdapat dalam fitur ini menggunakan metode *cosine similarity*. Secara umum, fungsi *similarity* adalah fungsi menerima dua buah objek dan mengembalikan nilai kemiripan antara kedua objek tersebut berupa bilangan riil [1].

Eska [2] menggunakan metode *Cosine Similarity* dalam Pengembangan Fitur Deteksi Dini Penyakit Pada Toko Online Obat Herbal sebagai pendeteksi dini penyakit sehingga didapatkan rekomendasi obat herbal sesuai dengan penyakit yang diderita. Aplikasi ini dibuat dengan meminta *input* dalam bentuk radio yang merepresentasikan jawaban "Ya" dan "Tidak" untuk setiap gejala yang ditanyakan.

Untuk meminimalisir jumlah pertanyaan pada proses *input* gejala, dilakukan pengelompokkan gejala dan pengeliminasian penyakit dalam beberapa tahap, sehingga gejala-gejala dari penyakit yang tidak mungkin diderita tidak akan ditanyakan lagi oleh aplikasi. Namun, fitur ini dapat membingungkan pengguna jika gejala yang dialami tidak terdapat pada daftar gejala yang ditanyakan.

Berdasarkan kondisi tersebut, maka diperlukan suatu aplikasi yang dapat mendeteksi hama dan penyakit dengan cara membuat daftar gejala-gejala yang dialami pengguna sebagai feedback. Sehingga nantinya fitur deteksi ini akan dibuat dengan berbasis tekstual, dimana pengguna akan memasukkan gejalanya berupa teks. Untuk mendapatkan informasi gejala yang dimaksud oleh user, perlu dilakukan proses pencarian gejala yang sesuai dengan gejala yang ada dalam fitur.

Tindakan, metode dan prosedur untuk menemukan kembali data yang tersimpan, kemudian menyediakan informasi mengenai subyek yang dibutuhkan didefinisikan sebagai Information Retrieval (IR) atau temu balik informasi (ISO 2382-1). Information Retrieval ini bertujuan untuk mencari dokumen-dokumen yang relevan dengan query user. Salah satu metode yang umumnya digunakan dalam bidang pencarian informasi

adalah metode Vector Space Model (VSM). Inti dari metode VSM adalah dasar dari tiap dokumen atau query diwakilkan oleh kata-kata yang terdapat di dalamnya (pengindeksan). Vektor yang terdiri dari kata-kata tersebut dapat didefinisikan untuk menggambarkan setiap bagian dari dokumen dan query, maka dokumen tersebut dapat ditentukan berhubungan dengan permintaan atau tidak berdasarkan hasil perhitungan korelasi antara mereka. Dokumen yang memiliki relativitas yang lebih besar dengan pencarian tertentu dianggap lebih terkait (Hongdan et al, 2011).

Untuk melakukan pencarian gejala-gejala hama dan penyakit dalam database yang relevan dengan input user, akan digunakan metode VSM yang hasilnya akan digunakan sebagai feedback dalam perhitungan deteksi hama dan penyakit selanjutnya. Pada penelitian ini, penerapan metode VSM dilakukan dengan menggambarkan gejala sebagai suatu dokumen, dimana gejala ini hanya terdiri dari beberapa suku kata dan berupa satu kalimat. Penentuan relevansi daftar gejala dengan input user dipandang sebagai pengukuran kesamaan (similarity measure) antara vektor gejala dengan vektor input. Semakin sama suatu vektor gejala dengan vektor input maka gejala tersebut dapat dipandang semakin relevan dengan masukan.

Pendeteksian hama dan penyakit dilakukan dengan membandingkan hasil perhitungan metode VSM tadi sebagai *feedback* pengguna dengan setiap gejala dari penyakit menggunakan metode *Cosine Similarity* yang memiliki nilai paling tinggi dengan interval 0-1. Dengan adanya fitur ini diharapkan pengguna dapat mendeteksi hama dan penyakit pada tanaman padi yang sedang dialami dengan mudah dan mendapatkan informasi cara penanganan dan pencegahannya.

2. DASAR TEORI

2.1 *Vector Space Model*

Vector Space Model adalah suatu model aljabar untuk mewakili dokumen teks sebagai suatu vektor pengenal, contohnya indeks kata. VSM biasanya digunakan dalam penyaringan informasi, temu balik informasi, pengindeksan, dan perankingan relevansi [3].

Pemikiran dasar dari metode VSM ini adalah merepresentasikan setiap kata independen dan setiap dokumen dinyatakan dalam sebuah vektor sehingga kompleksitas hubungan kata-kata menjadi sederhana dan dapat dihitung. Dalam VSM, setiap dokumen terdiri dari *term* (T1, T2, ..., Tn) dan setiap *term* Ti memiliki bobot Wi. *Term* (T1, T2, ..., Tn) dianggap sebagai salah satu elemen vektor dalam sistem koordinat N-dimensi [4].

TF-IDF merupakan sebuah skema pembobotan yang sering digunakan dalam VSM bersama dengan *cosine similarity* untuk menentukan kesamaan antara dua buah dokumen. TF-IDF mempertimbangkan frekuensi kata-kata yang berbeda dalam semua dokumen dan mampu membedakan dokumen. Dalam VSM, setiap vektor disusun oleh *term* dan bobot yang mewakili dokumen. Kesamaan dokumen dapat dinyatakan dengan sudut atau jarak antara vektor, semakin kecil sudut atau jarak berarti semakin mirip dua dokumen tersebut. TF merupakan *Term Frequency* dan IDF adalah *Inverse Document Frequency*. Rumusnya adalah sebagai berikut [5]:

$$W_{t,d} = TF_{t,d} * IDF_t \quad (1)$$

Keterangan :

$W_{t,d}$ = bobot dari *t (term)* dalam satu dokumen

$TF_{t,d}$ = frekuensi kemunculan *t (term)* dalam dokumen *d*

IDF_t = *Inverse document frequency*, dimana

$$IDF_t = \log\left(\frac{N}{n_t}\right) \quad (2)$$

Keterangan :

N = jumlah semua dokumen

n_t = jumlah dokumen yang mengandung *term t*

IDF mencerminkan penyebaran *term t* dalam keseluruhan dokumen sehingga dapat memperlihatkan perbedaan *term t* dalam tiap dokumen. TF mencerminkan penyebaran *term t* dalam sebuah dokumen. TF-IDF dapat membuat pengecualian bagi kata-kata yang berfrekuensi tinggi tetapi sedikit memiliki persamaan, sehingga TF-IDF merupakan algoritma yang efektif untuk perhitungan bobot *term t*.

Setelah pembobotan tiap *term* dilakukan, diperlukan perhitungan untuk melakukan perankingan ntuk mengukur kemiripan antara vektor *query* dan vektor dokumen yang akan dibandingkan. Salah satu metode yang biasa digunakan dalam perhitungan kemiripan adalah pengukuran *cosine*, yang menentukan sudut antara vektor dokumen dan vektor *query* dan didefinisikan sebagai :

$$Similarity\left(\vec{d_j}, \vec{q}\right) = \frac{\vec{d_j}.\vec{q}}{|\vec{d_j}|.|\vec{q}|} = \frac{\sum_{i=1}^{t}(w_{ij}\,.\,w_{iq})}{\sqrt{\sum_{i=1}^{t} w^2{}_{ij}\,.\sum_{i=1}^{t} w^2{}_{iq}}} \quad (3)$$

dimana $w_{q,t}$ adalah bobot dari *term t*, penyebut dalam persamaan ini disebut faktor normalisasi yang berfungsi untuk menghilangkan pengaruh panjang dokumen [5]. Normalisasi ini diperlukan karena dimana dokumen panjang akan cenderung memiliki nilai lebih besar karena memiliki frekuensi kemunculan kata yang besar pula.

Proses perankingan dari dokumen dapat dianggap sebagai proses pemilihan (vektor) dokumen yang dekat dengan (vektor) *query*, kedekatan ini diindikasikan dengan sudut yang dibentuk. Nilai *cosinus* yang cenderung besar mengindikasikan bahwa dokumen semakin sama dengan *query*. Nilai *cosinus* sama dengan 1 mengindikasikan bahwa dokumen sesuai dengan dengan *query* [6].

2.2 *Cosine Similarity*

Secara umum, fungsi *similarity* adalah fungsi yang menerima dua buah objek dan mengembalikan nilai kemiripan (*similarity*) antara kedua objek tersebut berupa bilangan riil. Umumnya, nilai yang dihasilkan oleh

fungsi *similarity* berkisar pada interval [0...1]. Namun ada juga beberapa fungsi *similarity* yang menghasilkan nilai yang berada di luar interval tersebut. Untuk memetakan hasil fungsi tersebut pada interval [0...1] dapat dilakukan normalisasi [1].

Cosine similarity adalah perhitungan kesamaan antara dua vektor n dimensi dengan mencari kosinus dari sudut diantara keduanya dan sering digunakan untuk membandingkan dokumen dalam text mining [8]. Rumus Cosine similarity adalah sebagai berikut:

$$Similarity\ (x,y) = \cos(\theta) = \frac{x.y}{\|x\|\|y\|} \quad (4)$$

Dimana :

x.y : *vector dot product* dari x dan y, dihitung dengan $\sum_{k=1}^{n} x_k y_k$ (5)

||x|| : panjang vektor x, dihitung dengan $\sum_{k=1}^{n} x_k^2$ (6)

$\|y\|$: panjang vektor y, dihitung dengan $\sum_{k=1}^{n} y_k^2$ (7)

Pang-Ning Tan [7] menjelaskan bahwa semakin besar hasil fungsi *similarity*, maka kedua objek yang dievaluasi dianggap semakin mirip. Jika sebaliknya, maka semakin kecil hasil fungsi *similarity*, maka kedua objek tersebut dianggap semakin berbeda. Pada fungsi yang menghasilkan nilai pada jangkauan [0...1], nilai 1 melambangkan kedua objek persis sama, sedangkan nilai 0 melambangkan kedua objek sama sekali berbeda.

3. METODOLOGI PENELITIAN

3.1 Studi Literatur dan Pemahaman

Penelitian ini dimulai dengan studi literatur untuk mengumpulkan bahan referensi yang membahas mengenai menghitung kemiripan antar dua teks atau dokumen menggunakan metode *Vector Space Model* dan *Cosine Similarity* guna memahami bagaimana proses serta cara penerapannya dalam hasil pendeteksian hama dan penyakit. Studi literatur ini mengambil dari jurnal-jurnal penelitian yang terkait dengan metode *Vector Space Model* dan metode *Cosine Similarity*.

3.2 Pengumpulan Data

Penelitian ini menggunakan data sekunder berupa data hama dan penyakit, gejala-gejalanya serta cara pencegahan dan penanganannya. Data diambil dari hasil penelitian berupa jurnal, buku yang diterbitkan oleh Puslitbang Tanaman Pangan berserta Balai Pengkajian Teknologi Pertanian dan IRRI, dan dari penjelasan dosen Fakultas Pertanian Universitas Sebelas Maret Surakarta.

3.3 Pemodelan Data

3.3.1 Proses Identifikasi input

Suatu penyakit dapat memiliki beberapa gejala. Gejala-gejala dari semua penyakit akan direpresentasikan sebagai dokumen dan diberi kode. Input pengguna bisa terdiri dari beberapa gejala, akan direpresentasikan sebagai *query*. Nantinya proses identifikasi input akan membandingkan kemiripan antar tiap gejala dari semua penyakit dengan *query* sehingga didapatkan daftar gejala yang dianggap *feedback* untuk proses menentukan penyakit.

Pada proses identifikasi input akan dicari kemiripan antara sebuah gejala dengan *query* (input pengguna). Kemiripan ini akan dihitung menggunakan metode *Vector Space Model*. VSM digunakan untuk membuat perankingan kemiripan antara gejala dan *query* setelah dilakukan pembobotan tiap kata atau *term.* Sebelum pembobotan dilakukan, data dokumen yang berupa gejala dan input data *query* akan melalui tahap-tahap *preprocessing*, kemudian setelah dilakukan pembobotan dihitung kemiripan antar *query* dan tiap gejala.

Pada hasil perhitungan VSM, akan diperoleh beberapa gejala yang mungkin dimaksud oleh pengguna dengan nilai akhir perhitungan *similarity* sebagai hasil perbandingan kemiripan gejala dengan *query*. Perankingan dilakukan dengan mengurutkan gejala yang memiliki nilai tertinggi sehingga gejala yang memiliki nilai paling tinggi akan dijadikan nilai *feedback* untuk digunakan dalam perhitungan identifikasi penyakit selanjutnya dengan diberi nilai biner 1 dan sisanya akan dianggap bernilai 0.

3.3.2 Proses Penemuan penyakit

Hasil *output* dari proses identifikasi input akan digunakan sebagai *feedback* untuk selanjutnya dihitung kemiripannya dengan gejala-gejala yang dimiliki oleh tiap penyakit sehingga akan diperoleh hasil deteksi beberapa penyakit yang mungkin dialami.

4. Pengembangan Aplikasi

Tahap implementasi dalam penelitian ini nantinya akan mengggabungkan metode *Vector Space Model* dan metode *cosine similarity* menggunakan bahasa pemrograman PHP dan MySQL sebagai *database server*.

4.1 Pengujian dan Analisa Hasil

Pada penelitian ini, pengujian akan dilakukan untuk mengukur ketepatan fitur deteksi. Pengujian akan dilakukan dua kali dengan perhitungan sebagai berikut :

1. Pengujian identifikasi *input*
   Pengujian identifikasi *input* dilakukan untuk mengukur akurasi fitur dalam membuat daftar gejala yang sesuai dengan *input*an *user*. Akurasi akan dihitung menggunakan rumus :
   Akurasi : $\frac{jumlah\ input\ benar}{jumlah\ input\ yang\ dimasukkan} \times 100\%$
2. Pengujian identifikasi *output*
   Pengujian identifikasi *output* dilakukan untuk mengukur akurasi fitur dalam memberi kemungkinan penyakit-penyakit yang sesuai dengan gejala yang sebelumnya sudah di*input*kan. Akurasi akan dihitung menggunakan rumus :
   Akurasi : $\frac{jumlah\ output\ benar}{jumlah\ percobaan\ yang\ dilakukan} \times 100\%$

5. HASIL DAN PEMBAHASAN

5.1 Gambaran Umum Fitur

Fitur deteksi ini akan digunakan sebagai fitur tambahan pada *website* informasi tentang tanaman padi. Pengguna nantinya akan menuliskan gejala-gejala yang dialami pada kolom yang tersedia. Untuk mengenali teks gejala yang dimasukkan oleh pengguna, nantinya masukan yang masih berupa teks dianggap sebagai *query* dan dihitung menggunakan metode *Vector Space Model* untuk menentukan gejala apa yang dimaksudkan oleh pengguna. Hasil akhir VSM inilah yang nantinya akan dianggap sebagai jawaban atau *feedback* dari pengguna untuk kemudian dicari penyakit yang sesuai hingga nantinya *output* akan berupa penyakit yang mungkin diderita beserta rekomendasi cara penanganan dan cara pencegahannya.

5.2 Pemodelan Data

5.2.1 Identifikasi Input

Data yang digunakan dalam penelitian ini terdiri dari 12 hama dan 11 penyakit tanaman padi dengan total jumlah gejala sebanyak 48 dalam 3 fase tanam. Untuk proses identifikasi *input*, 48 gejala yang ada akan melalui tahap *preprocessing* untuk dihitung kemiripan dengan *query*. Gejala yang ada direpresentasikan sebagai dokumen dengan kode G1, G2, ..., Gn dan *query user* juga akan dianggap sebagai dokumen pembanding dengan kode Q. Kemudian dilakukan tahap *preprocessing* sebagai berikut :

Menghapus format dan *markup* dalam gejala, kemudian dari tiap gejala dilakukan tokenisasi, yaitu memisahkan tiap kata dalam kalimat, selanjutnya dilakukan *filtering* untuk menghilangkan kata-kata sambung seperti “di”, “atau”, “seperti” dan lain-lain. Lalu tiap kata akan di-*stemming* dimana akan dihilangkan semua imbuhan yang melekat pada kata, seperti awalan dan akhiran.

Contoh :

G1 : Daun berwarna kuning
G2 : Daun berwarna putih
G3 : Batang berwarna kuning
Q : Daun kuning
Menjadi :
Daun, hijau, kuning, putih, dan batang.

Setelah didapatkan daftar *term*, kemudian dilakukan pembobotan dengan menghitung jumlah frekuensi kemunculan (tf) dan idf dengan rumus $IDF_t = \log\left(\frac{N}{d_f}\right)$, dimana N adalah jumlah keseluruhan gejala dan *query* dan df adalah jumlah gejala dan *query* yang memiliki *term* t.

Tabel 1. Perhitungan tf dan idf

Term	**Q**	**G1**	**G2**	**G3**	**IDF** $\left(\log\frac{N}{df}\right)$
Batang	0	0	0	1	$\log\left(\frac{4}{1}\right) = 0.60206$
Daun	1	1	1	0	$\log\left(\frac{4}{3}\right) = 0.12494$
Kuning	1	1	0	1	$\log\left(\frac{4}{3}\right) = 0.12494$
Putih	0	0	1	0	$\log\left(\frac{4}{1}\right) = 0.60206$
Warna	0	1	1	1	$\log\left(\frac{4}{3}\right) = 0.12494$

Kemudian dihitung bobotnya menggunakan rumus TFxIDF :

Tabel 2. Perhitungan pembobotan

Term	**Q**	**G1**	**G2**	**G3**
Batang	0	0	0	0.60206
Daun	0.12494	0.12494	0.12494	0
Kuning	0.12494	0.12494	0	0.12494
Putih	0	0	0.60206	0
Warna	0	0.12494	0.12494	0.12494

Kemudian dilakukan normalisasi sebagai berikut :

- Menghitung perkalian skalar antar bobot hitung dengan rumus $\sum_{t=1}^{v} w_{q,t} * w_{i,t}$ dan panjang vektor $\sqrt{\sum_{t=1}^{v} {w^2}_{q,t}}$

Tabel 3. Hasil perhitungan panjang vektor

Term	**Q**	**G1**	**G2**	**G3**
Batang	0	0	0	0.362476
Daun	0.01561	0.01561	0.01561	0
Kuning	0.01561	0.01561	0	0.01561
Putih	0	0	0.362476	0
Warna	0	0.01561	0.01561	0.01561
Jumlah	0.03122	0.04683	0.393696	0.393696
Akar	0.176692	0.216402	0.627452	0.627452

Hasil *cosine* $\left(\overrightarrow{G_t}, \vec{q}\right) = \frac{\overrightarrow{G_t}.\vec{q}}{|\overrightarrow{G_t}|.|\vec{q}|} = \frac{\sum_{i=1}^{t}(w_{ij}.w_{iq})}{\sqrt{\sum_{i=1}^{t} {w^2}_{ij} \cdot \sum_{i=1}^{t} {w^2}_{iq}}}$

$$G1 = \frac{0.03122}{0.176692 * 0.216402} = \frac{0.03122}{0.038237} = 0{,}816497$$

$$G2 = \frac{0.01561}{0.176692 * 0.627452} = \frac{0.01561}{0.110866} = 0{,}14080$$

$$G3 = \frac{0.01561}{0.176692 * 0.627452} = \frac{0.01561}{0.110866} = 0{,}14080$$

Dari hasil ini didapatkan bahwa gejala yang paling mewakili maksud dari *input user* yang berupa gejala daun kuning adalah G1 yaitu gejala daun berwarna kuning dengan nilai kemiripan sebesar 0,816497.

Tahap-tahap diatas akan dilakukan kembali sesuai dengan jumlah *input* yang dimasukkan oleh *user*.

5.2.2 Identifikasi penyakit

Hasil akhir perhitungan kemiripan antar *query* dan gejala akan dianggap sebagai *feedback.* Berikut ini adalah tahap-tahap identifikasi penyakit :

1. Melakukan transformasi biner, hasil perankingan dan penyakit yang ada ditransformasikan menjadi biner, dimana tiap gejala yang berupa hasil identifikasi *input* sebelumnya akan diberi nilai 1 dan yang lain diberi nilai biner 0.

Tabel 5. Contoh transformasi biner untuk *feedback* serta hama wereng coklat dan hama putih palsu

Hama atau penyakit	Gejala			
	Daun warna kuning	**Daun berwarna putih**	**Daun berwarna hijau tua**	**Batang berwarna kuning**
Hama wereng coklat	Ada	Tidak	Tidak	Ada
Hama wereng coklat (t)	1	0	0	1
Hama putih palsu	Tidak	Ada	Tidak	Tidak
Hama putih palsu (t)	0	1	0	0
Feedback	Ya	Tidak	Ya	Ya
Feedback (t)	1	0	1	1

Ket: (t) merupakan hasil tranformasi biner.

2. Menghitung nilai *cosine similarity* untuk deteksi penyakit.
   Contoh :
   $X = 1,0,0,1$ (hasil transformasi biner untuk hama wereng coklat)

   $Y = 1,0,1,1$ (hasil transformasi biner *feedback*)

   $$X.Y = (1*1) + (0*0) + (0*1) + (1*1)$$
   $$= 1 + 0 + 0 + 1 = 2$$
   $$\|x\| = \sqrt{1^2 + 0^2 + 0^2 + 1^2} = \sqrt{2} = 1{,}41421$$
   $$\|y\| = \sqrt{1^2 + 0^2 + 1^2 + 1^2} = \sqrt{3} = 1{,}73205$$
   $$cos(x,y) = \frac{x.y}{\|x\|.\|y\|} = \frac{2}{1{,}41421.1{,}73205}$$
   $$= \frac{2}{2{,}44729} = 0{,}816499$$

   Sehingga didapat hasil *similarity* antara hama wereng coklat dan *feedback* adalah sebesar 0,816499.

3. Pertanyaan mengenai masa tanaman
   Tanaman padi memiliki tiga fase dalam masa tanamnya, tiap fase memiliki beberapa penyakit yang sama. Maka dari itu untuk mempersempit kemungkinan kesalahan deteksi, selanjutnya akan ditanyakan pada fase apa gejala-gejala yang dialami sebelumnya.
4. Menampilkan hasil deteksi
   Tahap ini akan memberikan hasil hama atau penyakit apa yang sedang dialami oleh petani sesuai dengan fase yang dialami.

5.3 Hasil Pengujian

5.3.1 Pengujian Identifikasi Input

Pengujian identifikasi input dilakukan sebanyak 25 kali dengan 3 percobaan menggunakan 1 buah input, 19 percobaan menggunakan 2 buah input, dan 3 percobaan menggunakan 3 buah *input.* Pengujian ini dilakukan untuk mengukur akurasi ketepatan sistem dalam menghasilkan *output* yang sesuai dengan *query user*. Pengujian yang dilakukan terhadap keseluruhan 48 gejala menghasilkan akurasi identifikasi *input* sebesar 96%.

Kemudian pengujian identifikasi input dilakukan dengan menghilangkan beberapa input yang sama dan menghasilkan akurasi sebesar 98% dari total 22 input.

Pengujian identifikasi input yang dilakukan kembali dengan menggunakan varian kata seperti kalimat yang ada pada buku namun tetap mengacu pada data pengujian sebelumnya dengan pengeliminasian perulangan input menghasilkan akurasi sebesar 100% dari total 17 input.

5.3.2 Pengujian Identifikasi Output

Pada pengujian ini dilakukan 25 percobaan dengan menggunakan hasil identifikasi *input* sebelumnya. Pengujian ini dilakukan dengan cara membandingkan hasil dari *output* oleh sistem dengan pendapat para pakar. Dari pengujian yang dilakukan,, identifikasi *output* menghasilkan akurasi sebesar 92%.

Pengujian identifikasi hama dan penyakit kemudian dilakukan menggunakan input berupa kalimat yang diambil dari buku sebagai data pengujian baru dan menghasilkan akurasi sebesar 100%.

5.4 Analisis Hasil

Pengujian identifikasi *input* menunjukkan hasil *output* yang sesuai dengan nilai akurasi 96%. Dari 25 percobaan dengan total jumlah *input* sebanyak 50 didapatkan bahwa ada 2 hasil *output* dari proses identifikasi *input* yang tidak sesuai dengan yang dimaksud oleh *user*. Tabel 6 menunjukkan hasil *output* yang tidak sesuai dengan yang dimaksud oleh *user*.

Tabel 6. Hasil *output* yang tidak sesuai

Input user	***Output* identifikasi gejala**	**Nilai**	***Output* yang dimaksud**
Gabah bercak	Gabah coklat	0.52809	Gabah bercak bulat
	Gabah bercak bulat	0.47616	
	Gabah bercak berwarna coklat kemerahan	0.47341	

Kesalahan identifikasi input menggunakan metode VSM dipengaruhi oleh banyaknya suku kata pada satu gejala, dimana semakin banyak suku kata maka nilai *norm*

(panjang vektor) akan menjadi besar sehingga hasil similarity menjadi lebih kecil sehingga hasil perankingan *cosine similarity* pada VSM akan mengacu kepada gejala dengan panjang vektor yang lebih rendah.

Akurasi sebesar 92% dihasilkan dari kelanjutan identifikasi sebelumnya dengan *output* sesuai dengan pendapat pakar dan berada pada urutan pertama. Kegagalan fitur deteksi dalam mengidentifikasi hama dan penyakit tanaman padi disebabkan oleh kesalahan pada hasil *output* identifikasi gejala sebelumnya. Pengujian kemudian juga dilakukan menggunakan hasil identifikasi gejala menggunakan input berupa kalimat dari buku dengan fase yang sama dan menghasilkan *output* sesuai dengan deteksi pakar. Pengujian ini menghasilkan akurasi sebesar 100%. Hal ini menunjukkan bahwa metode *Cosine Similarity* dapat digunakan dalam identifikasi hama dan penyakit pada tanaman padi.

6. PENUTUP

Dari hasil penelitian dapat disimpulkan bahwa metode *Vector Space Model* dapat digunakan untuk melakukan identifikasi *input* dengan hasil gejala yang sesuai dengan *input user* sebagai *feedback* dan metode *Cosine Similarity* dapat digunakan untuk melakukan identifikasi *output* berupa hama atau penyakit padi yang sesuai, sehingga keduanya dapat dimanfaatkan untuk pendeteksian hama dan penyakit pada tanaman padi.

Pengujian identifikasi *input* menghasilkan akurasi sebesar 96% dengan dua hasil *output* tidak sesuai dan pengujian dilanjutkan ke tahap identifikasi hama dan penyakit padi dan menghasilkan akurasi sebesar 92%. Pengujian lain yang dilakukan untuk menguji ketepatan fitur dilakukan menggunakan input berupa kalimat menghasilkan akurasi sebesar 100%.

Dalam penelitian ini dapat diketahui bahwa banyaknya suku kata dalam satu gejala menjadi salah satu kekurangan metode VSM untuk ditemukannya gejala yang relevan dengan *query* pada saat proses identifikasi *input*. Kekurangan metode *Vector Space Model* dalam menentukan gejala yang relevan dengan kata kunci masih terjadi pada penelitian ini, maka untuk mengatasi kekurangan tersebut sebaiknya menggunakan metode *Vector Space Model* atau algoritma TF-IDF yang dimodifikasi untuk menekan pengaruh TF dan IDF yang tinggi.

Pencarian gejala awalnya dilakukan satu per satu untuk tiap *term* pada *query* dan dari semua gejala yang ada. Untuk meningkatkan relevansi penemuan gejala yang sesuai dapat dilakukan penghilangan *term* yang tidak berhubungan dengan *term* pada *query* sehingga gejala-gejala yang tidak memiliki *term* yang sama dengan *query* akan dieliminasi terlebih dahulu. Maka, identifikasi gejala dapat dilakukan beberapa cara yaitu dengan melakukan perhitungan terhadap masing-masing *term* pada *query* maupun kombinasi dari semua *term.*

Dalam proses identifikasi input, fitur tidak dapat memproses kesalahan pengetikan kata oleh *user* sehingga identifikasi input sebaiknya dikombinasikan dengan metode lain yang dapat mengatasi permasalahan tersebut.

Proses perhitungan nilai IDF pada saat proses pengindeksan selalu dihitung ulang sejak awal fitur digunakan, sehingga untuk meningkatkan efisiensi sebaiknya perhitungan TF-IDF dari gejala sudah ditentukan dan disimpan sehingga dapat diterapkan proses *cache* untuk perhitungan sehingga pendeteksian tidak perlu mengulang perhitungan untuk input yang sama dan sebelumnya pernah dimasukkan.

7. DAFTAR PUSTAKA

[1] Karhendana, A. (2008). *Pemanfaatan Document Clustering pada Agregator Berita*. Bandung: Program Studi Teknik Informatika ITB.

[2] Putra, E. S. R., Saptono, R., Wiharto. (2013). *Pengembangan Fitur Deteksi Dini Penyakit pada Toko Online Obat Herbal dengan Metode Cosine Similarity*. Universitas Sebelas Maret, Surakarta.

[3] Hongdan, et al. (2011). *A Document-Based Information Retrieval Model Vector Space*. IEEE. 65-68.

[4] Guo, Q. (2008). *The Similarity Computing of Document based on VSM*. IEEE.

[5] L. L. D., Chuang, H., Seamons, K. (1997). *Document Ranking and the Vector-Space Model.* IEEE. 67-79.

[6] Mandala, R., Setiawan, H. (2002). *Peningkatan Performansi Sistem Temu-Kembali Informasi dengan Perluasan Query Secara Otomatis*. Bandung: Institut Teknologi Bandung.

[7] Tan, P. N., Steinbach, M., Kumar, V. (2006). *Introduction to Data Mining*. London: Pearson Education Inc.

[8] Zhiqiang, L., Werimin, S., Zhenhua, Y. (2009). *Measuring Semantic Similarity between Words Using Wikipedia*. IEEE. 251-255

[9] Kowalski, G. J., Mark, T. M. (2007). *Information and Retrieval Systems (2nd ed)*. United States of America: Kluwer Academic Publisher.