

The Implementation of Jaro-Winkler Distance and Naive Bayes Classifier for Identification System of Pests and Diseases on Paddy

Andyta Permana Sari
Informatika, Fakultas MIPA
Universitas Sebelas Maret
Jl. Ir. Sutami No. 36A, Surakarta
andyta.permana@student.uns.ac.id

Ristu Saptono, S.Si.,M.T.
Informatika, Fakultas MIPA
Universitas Sebelas Maret
Jl. Ir. Sutami No. 36A, Surakarta
ristu.saptono@staff.uns.ac.id

Esti Suryani, S.Si, M.Kom
Informatika, Fakultas MIPA
Universitas Sebelas Maret
Jl. Ir. Sutami No.36A, Surakarta
suryapalapa@yahoo.com

ABSTRACT

The identification of pests and diseases on paddy are required to determine the control that can be done. Control of pests and diseases of paddy that are less precise as the excessive and prolonged use of pesticides can cause harm to the rice plant itself. This research aims to create a system to identify pests and diseases of paddy as well as provide alternative solutions to control it. This research proposes Jaro-Winkler Distance and Naive Bayes Classifier in identifying pests and diseases of paddy. The system takes text that contains symptoms of the disease as an input. The identification system in this research utilizes Jaro-Winkler Distance to identify symptoms from user input and Naive Bayes Classifier to identify what kind of pest or disease that infected the paddy. Developed applications got 93% accuracy, 94% precision, and 92% recall on evaluation using 10-fold cross-validation.

Keywords

Pests and Diseases of paddy, Jaro-Winkler Distance, Naive Bayes Classifier.

1. PENDAHULUAN

Penyebab utama rendahnya produksi padi sawah adalah banyaknya organisme pengganggu tanaman (OPT) yang dalam hal ini berupa hama dan penyakit tanaman [1]. Luas serangan hama dan penyakit padi menurut Pusat Data dan Sistem Informasi Pertanian tahun 2014, dalam kurun 2011 sampai dengan 2013 hama penggerek batang padi merusak sebesar 146.349 ha/tahun, wereng cokelat merusak sebesar 65.408 ha/tahun, tikus 37.956 ha/tahun, blast sebesar 56.193 ha/tahun, hawar daun sebesar 83.999 ha/tahun sedangkan tungro dapat merusak sebesar 10.250 ha/tahun [2].

Pemerintah telah mencanangkan program nasional pengendalian hama terpadu (PHT) untuk mengatasi masalah hama dan penyakit tanaman. Tujuan utama program ini adalah untuk mengurangi penggunaan pestisida sintetik yang berlebihan. Faktor kurangnya pengetahuan petani akan penggunaan pestisida yang benar, serta akibat yang ditimbulkannya baik bagi manusia maupun lingkungan menjadi penyebab tidak terkontrolnya penggunaan pestisida. Tindakan penggunaan pestisida yang dilakukan para petani umumnya dipengaruhi oleh faktor kebiasaan, kemudahan dalam penggunaan dan efektivitas [3]. Penggunaan pestisida yang berlebihan akan menyebabkan terjadinya degradasi lingkungan bahkan residu yang dihasilkan akan membahayakan manusia itu sendiri [4]. Penggunaan pestisida yang semakin lama akan mengakibatkan terjadinya resistensi pada hama itu sendiri [5].

Salah satu solusi untuk mengatasi masalah tersebut adalah memberikan sebuah sistem yang dapat menjadi alternatif panduan

bagi Petugas Penyuluh Lapangan (PPL) dan menambahkan pengetahuan bagi para petani dalam mengidentifikasi hama dan penyakit tanaman padi serta memberikan solusi alternatif penanggulangan yang tepat. Sistem identifikasi hama dan penyakit pada tanaman padi dalam penelitian ini menggunakan *input* tekstual sebagai masukan *user*, sehingga *user* dapat langsung memasukkan gejala yang dialami tanaman padi.

Penelitian ini bertujuan mengetahui hasil identifikasi hama dan penyakit tanaman padi dengan menerapkan metode *Jaro-Winkler Distance* untuk mengidentifikasi *input* gejala yang dimasukkan oleh *user*, serta *Naive Bayes Classifier* untuk mengklasifikasi jenis hama dan penyakit serta menampilkan solusi penanggulangan untuk tanaman padi mereka.

2. JARO-WINKLER DISTANCE

Jaro-Winkler distance adalah merupakan varian dari *Jaro distance* metrik yaitu sebuah algoritma untuk mengukur kesamaan antara dua *string*, biasanya algoritma ini digunakan di dalam pendeteksian duplikat atau kesamaan. Semakin tinggi nilai *Jaro-Winkler distance* untuk dua *string*, semakin mirip kedua *string* tersebut. Skor normalnya adalah 0 menandakan tidak ada kesamaan, dan 1 menandakan sama persis. [6]. Pada algoritma Jaro digunakan rumus untuk menghitung jarak (d_j) antara dua string yaitu S1 dan S2 dirumuskan pada Persamaan 1.

$$d_j = \frac{1}{3} \times \left(\frac{m}{|S_1|} + \frac{m}{|S_2|} + \frac{m-t}{m} \right) \quad (1)$$

Nilai m merupakan jumlah karakter yang sama persis dari kedua *string*. Suatu karakter dapat dikatakan memiliki kesamaan jika memiliki jarak teoritis yang tidak melebihi Persamaan 2. $|S_1|$ merupakan panjang *string* pertama sedangkan $|S_2|$ merupakan panjang dari *string* kedua dan t merupakan jumlah transposisi atau jumlah karakter yang tertukar posisi.

$$\left\lceil \frac{\max(|S_1|, |S_2|)}{2} \right\rceil - 1 \quad (2)$$

Jaro-Winkler distance menggunakan *prefix scale* (p) yang memberikan tingkat penilaian yang lebih, dan *prefix length* (l) yang menyatakan panjang awalan yaitu panjang karakter yang sama dari *string* yang dibandingkan sampai ditemukannya ketidaksamaan. Bila *string* S1 dan S2 yang diperbandingkan, maka *Jaro-Winkler distance*-nya (d_w) dirumuskan melalui Persamaan 3.

$$d_w = d_j + (l * p(1 - d_j)) \quad (3)$$

d_w merupakan nilai *jaro-winkler distance* dan d_j menunjukkan nilai *jaro distance* antara kedua string l menunjukkan panjang *prefix* umum di awal *string* dimana nilai maksimalnya 4 karakter atau panjang karakter yang sama sebelum ditemukan ketidaksamaan, sedangkan p konstanta *scaling factor*. Nilai standar untuk konstanta ini menurut Winkler adalah 0,1. [6]

3. NAIVE BAYES CLASSIFIER

Algoritma *Naive Bayes* didasarkan pada probabilitas bersyarat yang menggunakan teorema *Bayes*, yaitu sebuah formula yang dapat menghitung probabilitas suatu kejadian dengan menghitung frekuensi nilai-nilai dan kombinasi nilai-nilai dalam data historis [7]. Pada teorema bayes digunakan rumus untuk menghitung probabilitas peristiwa A terjadi setelah terjadi peristiwa B seperti terlihat pada Persamaan 4.

$$P(A|B) = \frac{P(A).P(B|A)}{P(B)} \quad (4)$$

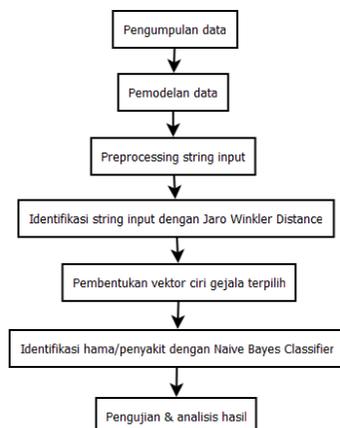
$P(A)$ menunjukkan probabilitas terjadinya peristiwa A sedangkan $P(B)$ menunjukkan probabilitas terjadinya peristiwa B. $P(A|B)$ merupakan probabilitas peristiwa A terjadi dengan syarat terjadinya peristiwa B sedangkan $P(B|A)$ menunjukkan Probabilitas peristiwa B terjadi dengan syarat terjadinya peristiwa A. Metode *Naive Bayes* untuk klasifikasi ditunjukkan pada Persamaan 5.

$$P(C|F_1 \dots F_n) = P(C) \prod_{i=1}^n P(F_i|C) \quad (5)$$

$P(C)$ adalah nilai probabilitas kelas dan $P(F_i|C)$ adalah nilai probabilitas kondisional setiap kelas. Keuntungan dari penggunaan *Naive Bayes Classifier* adalah algoritma ini tidak membutuhkan data *training* yang banyak untuk memperkirakan parameter yang diperlukan untuk klasifikasi [7]. Tak hanya itu, teorema *bayes* memiliki akurasi dan kecepatan yang sangat tinggi saat diaplikasikan ke dalam *database* dengan data yang besar [8].

4. METODOLOGI PENELITIAN

Gambar 1 menggambarkan metodologi identifikasi hama dan penyakit pada tanaman padi yang diusulkan dalam penelitian ini.



Gambar 1. Metodologi Penelitian

4.1 Pengumpulan Data

Data yang digunakan dalam penelitian ini berupa data sekunder yang terdiri dari 49 data gejala yang dialami tanaman, 23 jenis hama dan penyakit pada tanaman padi serta penanggulangan penyakit dan musuh alami dari hama tanaman padi. Data gejala serta hama dan penyakit padi yang digunakan dalam penelitian ini menggunakan *library* data yang ada pada penelitian [9]. Data hama dan penyakit yang menyerang tanaman padi yang digunakan dalam penelitian ini ditunjukkan pada Tabel 1. Data gejala yang dialami tanaman padi yang digunakan pada penelitian ini sejumlah 49 gejala. Beberapa contoh gejala yang dialami tanaman antara lain daun berwarna cokelat, daun menipis, daun mengering, gabah hampa, malai mati, batang pendek, jumlah anakan sedikit, dan lain sebagainya.

Tabel 1. Daftar Hama dan Penyakit

No	Nama Hama atau Penyakit
1	Wereng Cokelat
2	Wereng Hijau
3	Hama Putih Palsu
4	Tikus Sawah
5	Penggerek Batang Padi
6	Kepindang Tanah
7	Ulat Gerayak
8	Walang Sangit
9	Burung
10	Keong Mas
11	Ulat Tanah
12	Belalang
13	Bercak Daun Cercospora
14	Bercak Daun Cokelat
15	Blast
16	Hawar Pelelepah
17	Hawar Daun Bakteri
18	Hawar Daun Jingga
19	Busuk Batang
20	Busuk Pelelepah
21	Kerdil Rumpuk
22	Kerdil Hampa
23	Tungro

4.2 Pemodelan Data

Data gejala pada hama dan penyakit tanaman padi yang terkumpul, dilakukan pengelompokan domain. Domain merupakan bagian-bagian dari tanaman padi yang dibagi menjadi daun, batang, malai, gabah, anakan, pelepah, dan tanaman. Penentuan domain ini dilakukan agar mempermudah proses pengidentifikasian *input*. Sebagai contoh, untuk gejala “daun berwarna kuning” dikelompokkan pada domain “daun”, dan lain sebagainya.

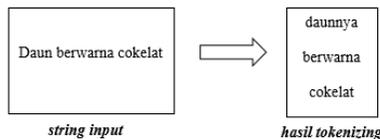
Data gejala pada hama dan penyakit tanaman padi juga dilakukan permodelan data untuk mengidentifikasi dan menganalisis hubungan untuk setiap hama dan penyakit dengan gejala yang dialami tanaman padi ke dalam bentuk biner 1 dan 0. Nilai biner “1” berarti gejala dialami hama atau penyakit dan nilai biner “0” berarti gejala tidak dialami hama atau penyakit.

Dari data yang telah ditransformasikan dari ke dalam bentuk biner tersebut diperoleh matriks berisi 249 data hubungan gejala dengan hama dan penyakit pada tanaman padi. Tahap selanjutnya adalah melakukan pembagian data *training* dan testing menggunakan

metode *k-fold cross validation* menjadi 10 bagian. Satu bagian dijadikan data *testing* dan bagian lainnya menjadi bagian *training*.

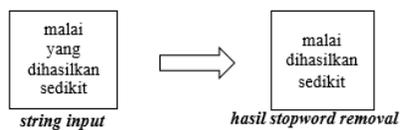
4.3 Preprocessing string input

Pada tahap ini, dilakukan *preprocessing* seluruh *input* gejala yang dimasukkan oleh *user* yang meliputi proses *tokenizing*, *stopword removal* dan *stemming*. Pada *input* gejala yang dimasukkan oleh *user* pertama kali dilakukan proses *tokenizing* untuk menghilangkan seluruh tanda baca pada teks *input* dan dilakukan pula pemotongan *string input* berdasarkan tiap kata yang menyusunnya. Gambar 1 menunjukkan contoh proses *tokenizing*.



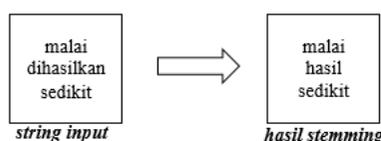
Gambar 2. Proses *Tokenizing*

Tahap selanjutnya adalah *stopword removal* yaitu penghilangan *useless word* seperti yang, dan, terdapat, hanya, dan lain-lain. Contoh masukan dan keluaran dari proses *stopword removal* dapat dilihat pada Gambar 2.



Gambar 3. Proses *Stopword Removal*

Tahap lanjutan jika masih terdapat ketidaksesuaian *string input* adalah proses *stemming*, yaitu mengubah *term* kata menjadi akar kata atau kata dasarnya. Proses *stemming* dilakukan dengan menggunakan algoritma Nazief & Adriani yang diambil dari *library* Sastrawi. Melalui *preprocessing* teks ini didapatkan masukan gejala yang sesuai dengan kebutuhan sistem. Contoh masukan dan keluaran dari proses *stemming* dapat dilihat pada Gambar 3.



Gambar 4. Proses *Stemming*

4.4 Identifikasi *string input* dengan *Jaro-Winkler Distance*

Dari hasil *preprocessing* teks *input* yang dimasukkan oleh *user* diperoleh masukan gejala yang sesuai dengan kebutuhan sistem. Masukan gejala dari *user* ini dilakukan penentuan domain terlebih dahulu dengan cara menghitung nilai kesamaan antara teks *input* dari pengguna dengan domain yang tersedia dengan metode *Jaro-Winkler Distance*. Domain yang memiliki nilai *Jaro-Winkler Distance* lebih dari 0.8 akan dijadikan acuan untuk mencocokkan kemiripan *input* gejala dengan *library* gejala pada *database*, sehingga ditemukan nilai *Jaro-Winkler Distance* dari setiap kalimat pada *library* gejala dengan *input* masukan *user*. Jika nilai *Jaro-Winkler Distance* tertinggi antara *input* dengan domain kurang dari *threshold* 0.8 maka sistem akan mencocokkan kemiripan *input*

gejala dengan seluruh gejala yang ada di *database*.

Nilai *Jaro-Winkler Distance* antara *input* gejala dengan tiap kata pada *library* gejala yang ada di *database* dilakukan pembobotan untuk setiap *library* gejala. Pembobotan ini dilakukan untuk mendapatkan tingkat kesamaan untuk setiap *library* gejala dengan *input* gejala dari *user*. Pembobotan ini dihitung dengan mengalikan nilai *Jaro-Winkler Distance* tertinggi dengan jumlah karakter pada setiap kata yang dibandingkan dan dibagi dengan jumlah total karakter dari setiap kata. Nilai pembobotan tertinggi dari perhitungan *Jaro-Winkler Distance* tertinggi digunakan sebagai gejala yang terdeteksi.

Jika nilai pembobotan tertinggi yang diperoleh dari perhitungan *Jaro-Winkler Distance* sebelumnya lebih dari satu gejala, dilakukan perhitungan nilai *Jaro-Winkler Distance* untuk setiap kalimat yang di *input*. Semakin banyak kata yang sama dengan gejala yang ada di *database* akan menghasilkan nilai *Jaro-Winkler Distance* yang semakin tinggi. Hasil nilai *Jaro-Winkler Distance* tertinggi dari setiap *library* gejala ini dianggap sebagai hasil identifikasi gejala dan selanjutnya ditransformasikan dalam bentuk biner 1 yang nantinya akan dihitung pada proses identifikasi hama dan penyakit dengan *Naive Bayes Classifier*.

4.5 Pembentukan vektor ciri gejala terpilih

Hasil gejala yang memiliki nilai pembobotan tertinggi dari proses yang telah dilakukan sebelumnya ditransformasikan ke dalam nilai biner 1 dan sisanya akan dianggap bernilai 0. Keseluruhan nilai biner yang didapat selanjutnya dijadikan vektor seperti yang digambarkan pada Tabel 2.

Tabel 2. Vektor Ciri Gejala Terpilih

Gejala 1	Gejala 2	Gejala ...	Gejala 49	
1	0	...	0	Penyakit 1
0	1	...	1	Penyakit 2
...	Penyakit ...
0	0	...	1	Penyakit23

4.6 Identifikasi penyakit dengan *Naive Bayes Classifier*

Pengidentifikasi hama atau penyakit dilakukan dengan membandingkan probabilitas terjadinya hama atau penyakit berdasarkan gejala yang dimasukkan *user* dengan metode *Naive Bayes Classifier* dengan langkah-langkah sebagai berikut:

1. Menghitung nilai *prior* setiap kelas penyakit.
2. Menghitung nilai *likelihood* atau probabilitas kondisional untuk setiap kelas penyakit.
3. Menghitung nilai posterior untuk setiap kelas penyakit.
4. Menentukan klasifikasi penyakit berdasarkan nilai posterior tertinggi.

Sistem akan menampilkan jenis hama atau penyakit yang memiliki probabilitas tertinggi sebagai prediksi penyakit yang telah menyerang tanaman padi.

4.7 Pengujian dan analisis hasil

Pembagian data *testing* dan data *training* pada penelitian ini menggunakan pengujian *k-fold cross validation* dengan $k=10$. Dari 249 data tersebut dibagi menjadi 10 bagian yaitu k_1 sampai dengan

k10. Untuk setiap bagian tersebut selanjutnya dilakukan evaluasi dan validasi hasil untuk mengetahui keakuratan hasil klasifikasi setiap bagian dan hasil akurasi keseluruhan. Evaluasi ini dilakukan dengan menggunakan *confusion matrix*. *Confusion matrix* adalah salah satu *tools* yang biasa digunakan dalam evaluasi *learning machine* yang memuat dua atau lebih kategori.

Tabel 3. Tabel Confusion Matriks

		Predicted Class	
		Penyakit 1	Penyakit 2
Actual Class	Penyakit 1	TP	FN
	Penyakit 2	FP	TN

True Positive (TP) dan *True Negative* (TN) adalah keadaan pada saat hasil *outcome* sesuai dengan kondisi sebenarnya yang terjadi. *False Positive* (FP) dan *False Negative* (FN) adalah keadaan dimana hasil *outcome* tidak sesuai dengan kondisi yang sebenarnya terjadi. Berdasarkan *confusion matrix* kemudian dilakukan penghitungan akurasi, *precision*, *recall*, dan *f-measure*.

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (7)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (8)$$

5. HASIL DAN PEMBAHASAN

5.1 Permodelan Data

Pada penelitian ini, diolah 249 data terdiri dari 12 hama dan 11 penyakit yang biasanya menyerang tanaman padi dimana total gejala secara keseluruhan berjumlah 49. Dari 249 data yang didapat dilakukan pembagian data menggunakan *k-folds cross validation* menjadi 10 bagian.

5.2 Preprocessing string input

Dalam penelitian ini pemisah untuk setiap gejala berupa tanda koma (.). Contoh masukan dan hasil identifikasi gejala setelah dilakukan pemisahan *input* dapat dilihat pada Tabel 4.

Tabel 4. Hasil Pemisahan Input

Input User	Identifikasi gejala setelah dilakukan pemisahan
Ada duannya yang menipis, tangkai malai patha, batang pendek, gabah hampa	1. Ada duannya yang menipis
	2. Tangkai malai patha
	3. Batang pendek
	4. Gabah hampa

Setiap *input* gejala yang telah dimasukkan *user* dilakukan *preprocessing* teks. Tahap *preprocessing* teks input ini menggunakan *library* dari Sastrawi. Setiap *input* ini dilakukan proses *stopword removal* untuk menghilangkan *stopword* atau kata sambung yang tidak diperlukan seperti dan, yang, pada, dan lain sebagainya. Proses ini menghasilkan *input* tanpa *useless word*. Selanjutnya setiap *input* diubah menjadi *token-token* atau *array* dan dilakukan penghapusan karakter selain huruf sehingga hanya menghasilkan *token* yang berisi kata. Setiap kata yang telah dilakukan proses *tokenisasi* tersebut dilakukan proses *stemming* yang menghasilkan *string* yang terdiri dari kata dasar dengan huruf

kecil secara keseluruhan. Tabel 5 menunjukkan tahapan *text processing* pada suatu *input* tekstual.

Tabel 5. Tahapan Text Preprocessing

Input tekstual	ada duannya yang menipis
Tokenizing	ada – duannya – yang – menipis
Stopword Removal	duannya – menipis
Stemming	duan - tipis

5.3 Identifikasi string input dengan Jaro-Winkler Distance

Proses identifikasi pertama yang dilakukan pada *string input* adalah dengan identifikasi domain tanaman padi dari *input* gejala yang dimasukkan oleh *user*. Domain tanaman padi pada penelitian ini dibagi menjadi tujuh antara lain daun, batang, gabah, malai, anakan, tanaman, dan pelepah. Berikut merupakan contoh perhitungan untuk mengidentifikasi domain dari *input* gejala yang dilakukan *user* dengan data gejala di *database* dimana pada contoh tersebut diberikan *input* dengan kesalahan penulisan (*typo*).

Input user : *gsbsh hampa*

Untuk setiap string dilakukan perhitungan nilai *jaro-winkler distance* dengan semua domain. Tabel 6 menunjukkan nilai *jaro-winkler distance* antara *string input* dengan domain.

Tabel 6. Jaro-Winkler Distance String Input dan Domain

domain \ string	gsbsh	hampa
daun	0	0,4833
batang	0,4556	0,5778
malai	0	0,6222
gabah	0,76	0,6
tanaman	0	0,5651
anakan	0	0,5778
pelepah	0,4476	0,5619

Berdasarkan Tabel 6 diperoleh nilai *jaro-winkler* tertinggi 0,76 dengan domain gabah. Domain yang memiliki nilai *Jaro-Winkler Distance* tertinggi dengan *string input* inilah yang akan dijadikan acuan untuk mencocokkan kemiripan masukan gejala dengan tiap kata pada *library* gejala yang ada di *database* yang memiliki domain yang telah teridentifikasi sebelumnya.

Langkah selanjutnya dilakukan perhitungan nilai *jaro-winkler distance* untuk setiap gejala dengan domain gabah. Masing-masing hasil perhitungan untuk setiap gejala dilakukan pembobotan dan diambil bobot tertinggi sebagai identifikasi *input* gejala yang dilakukan *user*, misalkan dihitung nilai *jaro-winkler distance* dengan gejala “gabah hampa”.

Input : *gsbsh hampa*

Gejala : *gabah hampa*

Tabel 7. Nilai Jaro-Winkler Distance antara Input dan Gejala

input \ gejala	gabah	hampa
gsbsh	0,76	0
hampa	0,6	1

$$\text{Akurasi} = \frac{231}{249} \times 100\% = 93\%$$

Dari *confusion* matriks pada Tabel 9 diperoleh hasil akurasi sebesar 93%, dengan nilai *precision* rata-rata sebesar 0.94, *recall* 0.92 dan *f-measure* sebesar 0.92. Dari hasil pengujian yang dilakukan, Tabel 4.13 menunjukkan masih terdapat nilai *recall* yang kecil yang menunjukkan masih adanya kesalahan prediksi yang berbeda dari data ahli seperti pada P07, P10, P12 dan P23.

Nilai *recall* kecil terjadi pada P10 (keong mas) sebesar 0.67. Dari *confusion* matriks pada Tabel 9, diketahui 3 dari 9 data terjadi kesalahan prediksi menjadi P19 (busuk batang). Kesalahan-kesalahan prediksi ini terjadi ketika *user* hanya memasukkan satu buah gejala yaitu “jumlah anakan sedikit”. Sistem mendeteksi menjadi P19 karena satu buah gejala tersebut memang dapat dialami oleh tanaman padi yang terkena hama wereng hijau (P02), keong mas (P10) dan penyakit busuk batang (P19).

Seperti halnya pada P12 (belalang) mendapatkan nilai *recall* kecil sebesar 0,5. Sebanyak 3 dari 7 data pada pengujian yang telah dilakukan mengalami kesalahan deteksi menjadi P11 (ulat tanah). Kesalahan prediksi ini juga terjadi ketika *user* hanya memasukkan satu buah gejala yaitu “tanaman hanya tinggal tulang-tulang daun”. Sistem mendeteksi menjadi P11 (ulat tanah) karena satu buah gejala tersebut memang dapat dialami oleh tanaman padi yang terkena hama ulat tanah (P11), dan hama belalang (P12).

Sebanyak 3 dari 11 data untuk pengujian P23 (tungro) juga mengalami kesalahan deteksi menjadi P02 (wereng hijau) sehingga mendapatkan nilai *recall* sebesar 0.73. Kesalahan-kesalahan prediksi ini terjadi ketika *user* memasukkan gejala “daun berwarna kuning” dan “jumlah anakan sedikit”. Sistem mendeteksi menjadi P02 (wereng hijau) karena kedua gejala tersebut memang dapat dialami oleh tanaman padi yang terkena hama wereng hijau (P02), dan penyakit tungro (P23).

Sebanyak 2 dari 9 data untuk pengujian P07 (ulat gerayak) juga mengalami kesalahan deteksi menjadi P11 (ulat tanah) sehingga mendapatkan nilai *recall* sebesar 0.78. Kesalahan-kesalahan prediksi ini terjadi ketika *user* memasukkan gejala “daun terdapat bercak putih” dan “tanaman hanya tinggal tulang-tulang daun”. Sistem mendeteksi menjadi P11 (ulat tanah) karena kedua gejala tersebut memang dapat dialami oleh tanaman padi yang terkena hama ulat gerayak (P07), dan ulat tanah (P11).

Berdasarkan analisa nilai *recall* untuk setiap kelas penyakit tersebut didapatkan kesimpulan bahwa masih terdapat beberapa kesalahan deteksi *output* ketika *user* hanya memasukkan gejala yang dapat terjadi pada beberapa hama dan penyakit. Penggunaan metode Naive Bayes Classifier untuk kondisi ini dirasa masih kurang akurat sehingga dapat ditambahkan metode *Association Rule* untuk mengetahui pola keterkaitan satu gejala dengan gejala lain.

Tabel 10. Nilai *Precision*, *Recall* dan *F-measure*

	<i>Precision</i>	<i>Recall</i>	<i>Fmeasure</i>
Wereng Cokelat	1	1	1
Wereng Hijau	0.75	0.82	0.78
Hama Putih Palsu	1	0.82	0.9
Tikus Sawah	1	1	1
Penggerak Batang Padi	1	0.92	0.96
Kepindang Tanah	1	1	1
Ulat Gerayak	1	0.78	0.875
Walang Sangit	1	1	1
Burung	1	1	1
Keong Mas	1	0.67	0.8
Ulat Tanah	0.53	0.89	0.67
Belalang	0.8	0.5	0.61
Bercak Daun Cercospora	1	1	1
Bercak Daun Cokelat	1	1	1
Blast	1	1	1
Hawar Pelepeh	1	1	1
Hawar Daun Bakteri	0.92	1	0.96
Hawar Daun Jingga	0.92	1	0.96
Busuk Batang	0.71	1	0.83
Busuk Pelepeh	1	1	1
Kerdil Rumput	1	1	1
Kerdil Hampa	1	1	1
Tungro	1	0.73	0.84
Rata-Rata	0.94	0.92	0.92

6. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan dapat disimpulkan bahwa metode *Jaro-Winkler Distance* dan *Naive Bayes Classifier* dapat digunakan untuk melakukan identifikasi hama dan penyakit pada tanaman padi. Dari sepuluh *dataset* pengujian *output* menggunakan *confusion* matriks yang telah dilakukan, diperoleh nilai akurasi keseluruhan dari sepuluh pengujian sebesar 93%, dengan nilai *precision* rata-rata sebesar 0.94, nilai *recall* sebesar 0.918 dan *f-measure* sebesar 0.921.

Umumnya kesalahan deteksi *output* pada penelitian ini terjadi ketika *user* hanya memasukkan gejala yang dapat terjadi pada beberapa hama dan penyakit. Penggunaan metode *Naive Bayes Classifier* untuk kondisi ini dirasa masih kurang akurat sehingga dapat ditambahkan metode *Association Rule* untuk mengetahui pola keterkaitan satu gejala dengan gejala lain.

Dari penelitian ini, masih terdapat beberapa potensi penelitian lanjutan yang dapat dilakukan untuk sistem identifikasi hama dan penyakit padi. Adapun saran yang dipertimbangkan untuk pengembangan penelitian ini selanjutnya adalah penambahan metode *Association Rule* untuk mengetahui pola keterkaitan satu gejala dengan gejala lain sehingga hasil *output* ketika *user* hanya memasukkan satu buah gejala dapat lebih akurat, penambahan fungsi sinonim kata dalam *database* gejala yang dialami tanaman padi, sehingga penggunaan kata yang memiliki arti yang sama dapat teridentifikasi dengan benar. Penambahan *autotext input* sehingga hasil akurasi *input* gejala dan hasil deteksi hama dan penyakit lebih akurat.

7. DAFTAR PUSTAKA

- [1] Wilyus, W., Ratna, Y., & Yunita, W., "Pengendalian Hama Terpadu (PHT) Biointensif pada Tanaman Padi di Desa Senaning", *Jurnal Karya Abdi Masyarakat*, 35-42, 2017.
- [2] Gultom, R., Hasanah, L., & Supriyatna, M. A., "Statistik Iklim, Organisme Pengganggu Tanaman dan Dampak Perubahan Iklim", *Pusat Data dan Sistem Informasi Pertanian Sekretariat Jenderal Kementerian Pertanian*, Jakarta, 2014.
- [3] Apriliantina, R. S., "Pengetahuan, Sikap dan Tindakan Petani dalam Penggunaan Pestisida di Kabupaten Indramayu Jawa Barat", 2013.
- [4] Prayito, W., Saam, Z., Nurhidayah, T., "Hubungan Pengetahuan, Persepsi dan Perilaku Petani dalam Penggunaan Pestisida pada Lingkungan di Kelurahan Maharatu Kota Pekanbaru", *Pusat Penelitian Lingkungan Hidup Universitas Riau*, 2014.
- [5] Tjahjadi, N., "Hama dan Penyakit Tanaman", Kanisius, Yogyakarta, 1989.
- [6] Kurniawati, A., Puspitodjati, S., & Rahman, S., "Implementasi Algoritma Jaro-Winkler Distance untuk Membandingkan Kesamaan Dokumen Berbahasa Indonesia", *Proceeding Seminar Ilmiah Nasional ke-5 KOMMIT*, Bali, 2010.
- [7] Parthiban, G., Rajesh, A., & Srivatsa, S., "Diagnosis of Heart Disease for Diabetic Patients using Naive Bayes Method", *International Journal of Computer Application*, 7-11, 2011.
- [8] Purnana, P., & Supriyanto, C., "Deteksi Penyakit Diabetes Type II dengan Naive Bayes Berbasis Particle Swarm Optimization", *Jurnal Teknologi Informasi*, 49-53, 2013.
- [9] Rahadian, M., Saptono, R., & Doewes, A., "Deteksi Hama dan Penyakit Tanaman Padi Memanfaatkan Input Tekstual dengan Metode Cosine Similarity", *Prosiding Seminar Ilmiah Ilmu Komputer 2014*, 39-54, 2014.