

PURWARUPA SISTEM PAKAR IDENTIFIKASI JAMUR LAYAK KONSUMSI BERBASIS WEB

Agung Wibowo

Teknik Informatika/STMIK Nusa Mandiri Sukabumi
Jl. Veteran II No.20A Kota Sukabumi, 43114 Indonesia
agung.awo@nusamandiri.ac.id

Page | 112

Abstrak — Jamur berdasarkan manfaatnya dapat digolongkan menjadi layak makan, beracun dan obat. Penelitian dan aplikasi indentifikasi jamur layak makan/konsumsi masih sangat sedikit padahal Indonesia yang secara geografis berada di daerah tropis mendukung pertumbuhan jamur. Sistem pakar yang dikembangkan pada penelitian ini didasarkan hasil pengujian dataset jamur family Agaricus dan Lepiota menggunakan algoritma klasifikasi *decision tree* (J48), dari hasil pengujian rerata galat menggunakan *10-fold cross validation* diperoleh nilai galat 0 yang artinya algoritma ini dapat meng-klasifikasi jamur layak konsumsi dengan akurasi sebesar 100%. Paper ini juga menyertakan algoritma dan sumber kode program berbasis web yang dapat dimodifikasi dan diterapkan pada media/bahasa pemrograman lain.

Keywords — Jamur, Prunning, Decision Tree, J48, Data Mining.

I. PENDAHULUAN

Indonesia di daerah tropis dan memiliki banyak hutan hujan tropis cukup mendukung pertumbuhan jamur [1], dan habitat jamur ada pada bermacam lingkungan dan berasosiasi dengan banyak organisme[2]. Jamur sendiri tidak dapat tumbuh sepanjang tahun, karena hidup jamur sangat tergantung pada suhu dan kelembaban[3]. jamur dapat dibedakan berdasarkan bentuk payung (mangkuk, kuping, setengah lingkaran dan bulat), akar semu, batang/tangkai (*stipe*), tudung/cap, bilah/*gills*, cincin(*annulus/ring*) dan cawan(*volva*)[3]. Penelitian mengenai jamur di Indonesia masih sangat terbatas [4],[5], berdasarkan manfaatnya jamur dapat digolongkan menjadi tiga: dapat dikonsumsi, sebagai obat dan beracun. Berdasarkan bentuknya, sukar diketahui ciri suatu jamur beracun atau tidak [6]. Hasil pencarian menggunakan google cendikia per juni 2017 ada dua peneliti yang membahas identifikasi jamur [7][8]tetapi belum ada yang mempublikasikan sistem pakar indentifikasi jamur.

Algoritma yang digunakan dalam aplikasi ini adalah algoritma C4.5 atau lebih dikenal dengan *Decision tree* yang sebelumnya telah melalui pengujian menggunakan berdasarkan data publik dari UCI *Machine Learning repository*.

Aplikasi sistem pakar pada umumnya dikembangkan menggunakan tiga media: personal *Personal Computer* (PC)*desktop*, *Web based* dan *mobile application*, dari ketiganya yang paling fleksibel dari sisi pengembangan dan pengguna adalah *web based* karena beberapa hal berikut, yang pertama adalah mudah digunakan dibandingkan

dengan perangkat lunak pengembang, kita tidak harus *download* dan menginstall *driver* tambahan, tidak harus *update* versi, tidak harus khawatir masalah *hardware* dan kompatibilitas yang tidak mendukung; yang kedua aman, tidak ada data yang disimpan atau diambil dari perangkat lokal pengguna, data sensitif seluruhnya disimpan di *server*; dan yang ketiga, dapat dijalankan dimana saja, bahkan pada perangkat bergerak [9].

Bahasa pemrograman paling populer yang digunakan untuk mengembangkan aplikasi berbasis web adalah ASP.NET, C, C++, D, Erlang, Go, Hack, Java, JavaScript, Perl, PHP, Python, Ruby, Scala dan Xhp. Bahasa pemrograman berbasis web yang paling digunakan pada situs populer (Google.com, Facebook.com, youtube.com, Yahoo, Amazon.com, Wikipedia.com, Twitter.com, Bing, eBay.com, MSN.com, Microsoft, LinkedIn.com, Pinterest dan Wordpress.com) adalah JavaScript[10], Javascript merupakan bahasa pemrograman yang dirilis tahun 1995, javascript berbeda dengan bahasa pemrograman java. Javascript adalah bahasa naskah yang dibaca dan dieksekusi menggunakan interpreter dan syntax serta strukturnya mengacu pada bahasa pemrograman C. javascript juga mendukung *object oriented programming* [11]sehingga aplikasi ini dibuat sebagai bentuk penerapan model algoritma pengidentifikasi jamur layak konsumsi berupa purwarupa sistem pakar berbasis web menggunakan javascript.

Pengolah dataset pada penelitian ini menggunakan WEKA, WEKA (Waikato Environment for Knowledge analysis) adalah suatu perangkat lunak pembelajaran mesin yang populer ditulis dengan java, yang dikembangkan di

Universitas Waikato di Selandia Baru. WEKA adalah perangkat lunak gratis yang tersedia di bawah GNU General Public Licence. WEKA menyediakan penggunaan teknik klasifikasi menggunakan decision tree dengan algoritma J48. Teknik klasifikasi dan algoritma yang digunakan di WEKA disebut classifier[11].

II. METODE PENELITIAN

Data yang digunakan pada aplikasi ini menggunakan data sekunder yang bersumber dari UCI Machine Learning repository dan bersifat terbuka.

A. Decision Tree

Semua ahli teori dan spesialis terus-menerus mencari teknik untuk membuat proses lebih efisien, hemat biaya dan akurat. Decision Tree adalah sistem pendukung keputusan yang menggunakan keputusan berbentuk grafik mirip pohon dan kemungkinan hasilnya, termasuk probabilitas hasil, biaya sumber daya, dan utilitas. Decision tree, atau disebut juga pohon klasifikasi, digunakan untuk mempelajari fungsi klasifikasi yang menyimpulkan nilai atribut (variabel) dependen dengan nilai atribut (variabel) independen. Decision tree adalah metode pendekatan yang paling kuat dalam penemuan pengetahuan dan data mining. Metode ini mencakup teknologi penelitian data besar dan kompleks untuk bisa menemukan pola yang bermanfaat. Ide dari decision tree sangat penting karena memungkinkan pemodelan dan meng-ekstraksi pengetahuan dari sebagian besar data yang ada[12].

B. Pruning Tree

Terdapat dua cara pruning yaitu postpruning dan prepruning. Postpruning, pemangkasan dilakukan setelah pohon keputusan jadi, dengan mengganti subtree menjadi leafnode. Sedangkan prepruning, pemangkasan dilakukan selama pohon keputusan dikonstruksi, kapan harus berhenti dan tidak melanjutkan pembentukan subtree.

Prepruning

Prepruning memiliki keuntungan lebih hemat waktu. Kita tidak membuang waktu untuk membuat subtree yang pada akhirnya tidak digunakan (setelah dilakukan simplifikasi). Prepruning dilakukan dengan melakukan statistical significance test pada setiap splitting. Jika hasil tesnya dibawah threshold artinya konstruksi pohon keputusan berhenti. Menentukan threshold ini yang tidak mudah, karena jika terlalu tinggi maka pohon keputusan yang dihasilkan terlalu simple dan prematur (underfit) dan jika threshold terlalu rendah maka simplifikasinya terlalu kecil (overfit). Umumnya statistical significant test yang diterapkan ialah menggunakan chi-squared test.

Postpruning

Kebanyakan algoritma decision tree menerapkan postpruning, walaupun membutuhkan waktu lebih banyak. Postpruning dapat menghasilkan simplified tree yang lebih baik akurasi. Dua cara yang dapat dilakukan dalam melakukan postpruning.

- Subtree replacement

Subtree replacement adalah mengganti subtree dengan sebuah leafnode. Replacement berjalan mulai dari node yang paling bawah atau yang memiliki leaf node hingga root node.

- Subtree raising

Pada subtree raising, sebuah node akan dibuang dan digantikan dengan salah satu cabang node tersebut[14].

Pruning menjadi salah satu cara untuk merampingkan struktur dari decision tree sehingga proses generalisasi datanya menjadi lebih baik, menghasilkan prediksi yang lebih akurat dan memudahkan interpretasinya. Dengan ini kita dapat menghindari terjadinya overfitting. Ada beberapa metode Pruning, salah satunya digunakan dalam algoritma C4.5 (J48 dalam WEKA). Metode pruning yang digunakan dalam algoritma ini dinamakan pessimistic pruning/prepruning yang didasarkan koreksi statistik. Selain itu, algoritma ini hanya melakukan training set untuk membentuk pohon keputusan dan tidak membutuhkan validation set atau test set untuk mengukur galat selama proses pruning. Pengukuran rerata galat cukup dilakukan menggunakan upper confidence limit (one-tail), yang dihitung menggunakan rumus berikut:

$$e = \frac{f + \frac{z^2}{2N} + z \sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}}}{1 + \frac{z^2}{N}}$$

Dimana N jumlah sample, $f = \frac{E}{N}$, E adalah jumlah galat, dan z adalah z-score untuk interval kepercayaan (confidence interval) c. secara default, C4.5 akan menggunakan 25% confidence interval (anda dapat mengetahui nilainya dengan bantuan tabel z). [13]

Data set bernama mushroom dengan 8124 rekord dan 22 atribut yang terbagi kedalam katagori layak konsumsi berjumlah 4208 rekord (51.8%) dan Decision Tree sisanya 3916 rekord (48.2%) beracun dari 23 spesies jamur pada family Agaricus dan Lepiota. Keterangan dari data set jamur berupa 22 buah informasi atribut.

Attribute Information:

- (classes: edible=e, poisonous=p)
1. cap-shape: bell=b, conical=c, convex=x, flat=f, knobbed=k, sunken=s
 2. cap-surface: fibrous=f, grooves=g, scaly=y, smooth=s
 3. cap-color: brown=n, buff=b, cinnamon=c, gray=g, green=r, pink=p, purple=u, red=e, white=w, yellow=y
 4. bruises?: bruises=t, no=f
 5. odor: almond=a, anise=l, creosote=c, fishy=y, foul=f, musty=m, none=n, pungent=p, spicy=s
 6. gill-attachment: attached=a, descending=d, free=f, notched=n
 7. gill-spacing: close=c, crowded=w, distant=d
 8. gill-size: broad=b, narrow=n
 9. gill-color: black=k, brown=n, buff=b, chocolate=h, gray=g, green=r, orange=o, pink=p, purple=u, red=e, white=w, yellow=y
 10. stalk-shape: enlarging=e, tapering=t
 11. stalk-root: bulbous=b, club=c, cup=u, equal=e, rhizomorphs=z, rooted=r, missing=?
 12. stalk-surface-above-ring: fibrous=f, scaly=y, silky=k, smooth=s
 13. stalk-surface-below-ring: fibrous=f, scaly=y, silky=k, smooth=s
 14. stalk-color-above-ring: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
 15. stalk-color-below-ring: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
 16. veil-type: partial=p, universal=u
 17. veil-color: brown=n, orange=o, white=w, yellow=y
 18. ring-number: none=n, one=o, two=t
 19. ring-type: cobwebby=c, evanescent=e, flaring=f, large=l, none=n, pendant=p, sheathing=s, zone=z
 20. spore-print-color: black=k, brown=n, buff=b, chocolate=h, green=r, orange=o, purple=u, white=w, yellow=y

21. population: abundant=a, clustered=c, numerous=n, scattered=s, several=v, solitary=y
22. habitat: grasses=g, leaves=l, meadows=m, paths=p, urban=u, waste=w, woods=d

WEKA digunakan sebagai alat bantu pengujian dan pembentukan *rule set (decision tree)* dari data set yang diujikan. Pengujian rerata galat data menggunakan teknik pembelajaran *10-fold cross validation* sebanyak 10 kali perulangan, karena menurut [11] pengujian *10-fold cross validation* yang hanya dilakukan hanya sekali, nilai galatnya tidaklah dapat diandalkan, pengujian berulang pada data set yang sama seringkali menampilkan nilai galat yang berbeda.

III. HASIL DAN PEMBAHASAN

Hasil dari 10 kali *10-fold cross validation* diperoleh hasil akurasi menggunakan algoritma Decision Tree sebesar 100% dan rerata waktu olah adalah 0,14 detik, hasil perhitungan galat dapat dilihat pada table 1.

TABEL 1
HASIL HITUNG GALAT

	Stratified cross-validation	Summary
Correctly Classified Instances	8124	100 %
Incorrectly Classified Instances	0	0%
Kappa statistic	1	
Mean absolute error	0	
Root mean squared error	0	
Relative absolute error	0	
Root relative squared error	0	
Total Number of Instances	8124	

Terninologi pengujian statistik:

1. *Kappa Statistic* adalah matrik yang membandingkan akurasi teramati dengan akurasi yang diharapkan (peluang acak)
2. *Mean Absolute Error*: rerata absolut galat antara nilai yang diamati dengan yang diprediksi.
3. *Root Mean Squared Error*: mengukur perbedaan antara nilai (sampel dan nilai populasi) yang diprediksi oleh model atau estimator dengan nilai yang benar-benar diamati

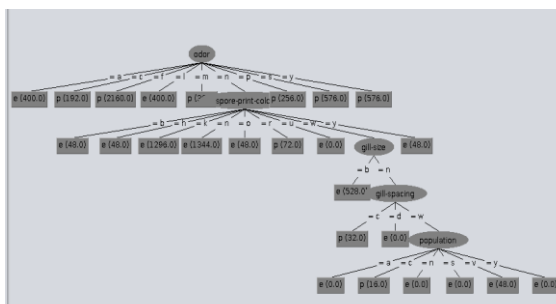
4. *Relative Absolute Error*: Rasio absolute galat pengukuran terhadap nilai pengukuran yang diterima

Bentuk dari pruning *decision tree* nya akan menghasilkan 25 buah daun dan 30 cabang yang dijabarkan sebagai berikut,

```

odor = a: e (400.0)
odor = c: p (192.0)
odor = f: p (2160.0)
odor = l: e (400.0)
odor = m: p (36.0)
odor = n
| spore-print-color = b: e (48.0)
| spore-print-color = h: e (48.0)
| spore-print-color = k: e (1296.0)
| spore-print-color = n: e (1344.0)
| spore-print-color = o: e (48.0)
| spore-print-color = r: p (72.0)
| spore-print-color = u: e (0.0)
| spore-print-color = w
| | gill-size = b: e (528.0)
| | gill-size = n
| | | gill-spacing = c: p (32.0)
| | | gill-spacing = d: e (0.0)
| | | gill-spacing = w
| | | | population = a: e (0.0)
| | | | population = c: p (16.0)
| | | | population = n: e (0.0)
| | | | population = s: e (0.0)
| | | | population = v: e (48.0)
| | | | population = y: e (0.0)
| spore-print-color = y: e (48.0)
odor = p: p (256.0)
odor = s: p (576.0)
odor = y: p (576.0)
    
```

Bentuk visual dari *decision tree* diatas dapat dilihat pada gambar 1.



Gambar 1. *Decision tree* identifikasi jamur layak konsumsi

A. Algoritma

Seluruh dokumen harus dalam Times New Roman atau Times font. Type 3 font tidak boleh digunakan. Jenis font lain dapat digunakan jika diperlukan untuk tujuan khusus. Fitur ukuran font terlihat pada Tabel 1.

```

IF odor = a || l THEN
Layak_Konsumsi
IF odor = c || f || m || p || s ||
y THEN Beracun
IF odor = n THEN
    IF Spore_Print_Color = r
    THEN Beracun
    IF Spore_Print_Color = b || h ||
k || n || o || u THEN
        Layak_Konsumsi
    IF Spore_Print_Color = w THEN
        IF Gill_Size = b THEN Beracun
        IF Gill_size = n THEN
            IF Gill-Spacing = c THEN
Beracun
                IF Gill-Spacing = d THEN
Layak_Konsumsi
                    IF Gill-Spacing = w THEN
                        IF Population = c THEN
Beracun
                            ELSE Layak_Konsumsi
    
```

B. Kode Program

Kode Program berikut juga dapat diakses secara online pada URL <http://jsfiddle.net/7Shsy/330/>

```

<select id="opts"
onchange="showForm()" > <option
value="0">Odor</option>
<option value="1">a/l</option>
<option
value="2">c/f/m/p/s/y</option>
<option value="3">n</option>
</select>

<div id="f1" style="display:none">
<form name="form1">
<select id="opts2"
onchange="showForm()" >
<option value="0">Spore Print
Color</option>
<option value="1">r</option>
<option
value="2">b/h/k/n/o/u</option>
<option value="3">w</option>
</select>
</form>
</div>

<div id="f2" style="display:none">
<form name="form1">
<select id="opts3"
onchange="showForm()" >
<option value="0">Gill Size</option>
<option value="1">b</option>
<option value="2">n</option><
/option>
</select>
</form>
</div>
    
```

```
<div id="f3" style="display:none">
<form name="form1">
<select id="opts4"
onchange="showForm()">
<option value="0">Gill
Spacing</option>
<option value="1">c</option>
<option value="2">d</option>
<option value="3">w</option>
</select>
</form>
</div>
```

```
<form name="form1">
<select id="opts5"
onchange="showForm()">
<option
value="0">Population</option>
<option value="1">c</option>
<option value="2">a/n/s/v/y</option>
</select>
</form>
</div>
```

```
<div id="f4" style="display:none">
```

```
<script type="text/javascript">
function showForm()
{
var selopt =
document.getElementById("opts").val
ue;
if (selopt == 1)
{
window.alert("Edible");
document.getElementById("f1").style
.display = "none";
}
if (selopt == 2)
{
window.alert("Poison");
document.getElementById("f1").style
.display = "none";
}
if (selopt == 3)
{
document.getElementById("f1").style
.display = "block";
}
if (selopt == 0)
{
location.reload();
}
}
```

```
var selopt =
document.getElementById("opts2").va
lue;
if (selopt == 1)
```

```
{ window.alert("Poison");
document.getElementById("f2").style
.display = "none";
}
if (selopt == 2)
{
window.alert("Edible");
document.getElementById("f2").style
.display = "none";
}
if (selopt == 3)
{ document.getElementById("f2").sty
le.display = "block";
}
if (selopt == 0)
{ document.getElementById("f2").sty
le.display = "none";
}
}
```

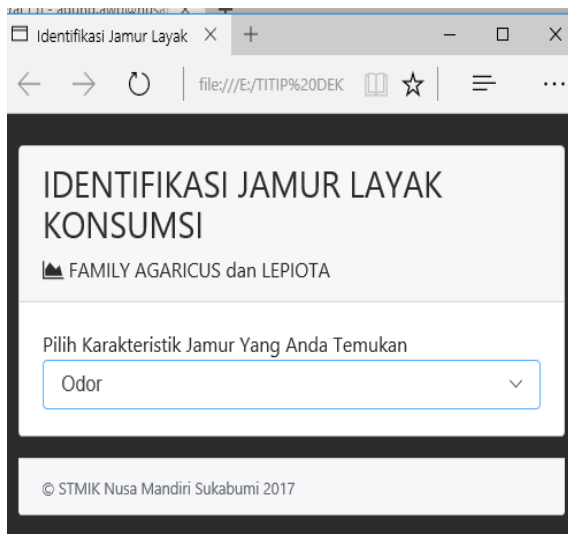
```
var selopt =
document.getElementById("opts3").va
lue; if (selopt == 1)
{
window.alert("Edible");
document.getElementById("f3").style
.display = "none";
}
if (selopt == 2)
{
document.getElementById("f3").style
.display = "block";
}
if (selopt == 0)
{
document.getElementById("f3").style
.display = "none";
}
}
```

```
var selopt =
document.getElementById("opts4").va
lue; if (selopt == 1)
{
window.alert("Poison");
document.getElementById("f4").style
.display = "none";
}
if (selopt == 2)
{
window.alert("Edible");
document.getElementById("f4").style
.display = "none";
}
if (selopt == 3) {
document.getElementById("f4").style
.display = "block";
}
}
if (selopt == 0)
{
```

```
document.getElementById("f4").style  
.display = "none";  
}  
}  
</script>
```

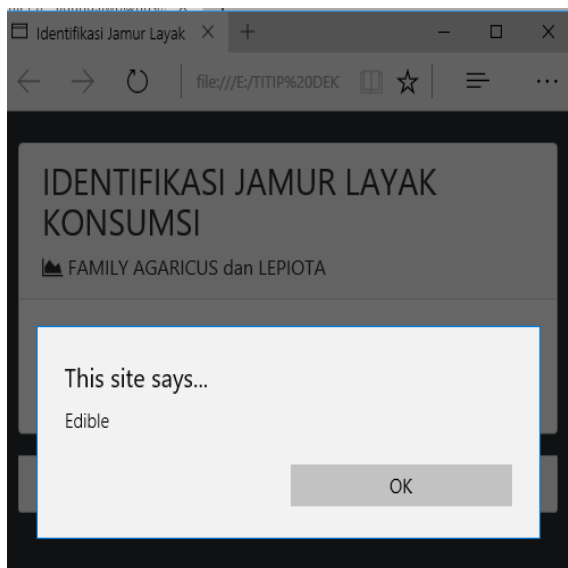
C. Tampilan Aplikasi

Menu awal identifikasi jamur, pengguna akan diminta menentukan aroma /bau (*Odor*) dari jamur yang akan diidentifikasi seperti yang terlihat pada gambar 2.

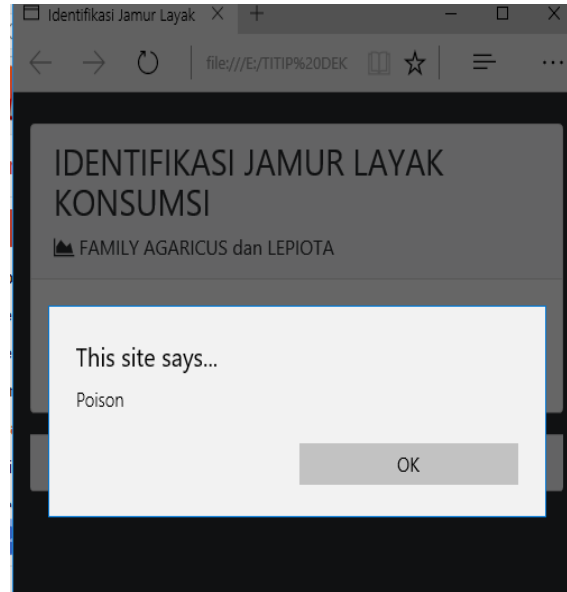


Gambar 2. Menu utama identifikasi jamur

Hasil identifikasi akan ditampilkan oleh sistem dalam bentuk *alert windows*, untuk hasil identifikasi jamur layak konsumsi akan terlihat seperti pada gambar 3 dan hasil identifikasi jamur beracun akan ditampilkan *alert window* seperti pada gambar 4.

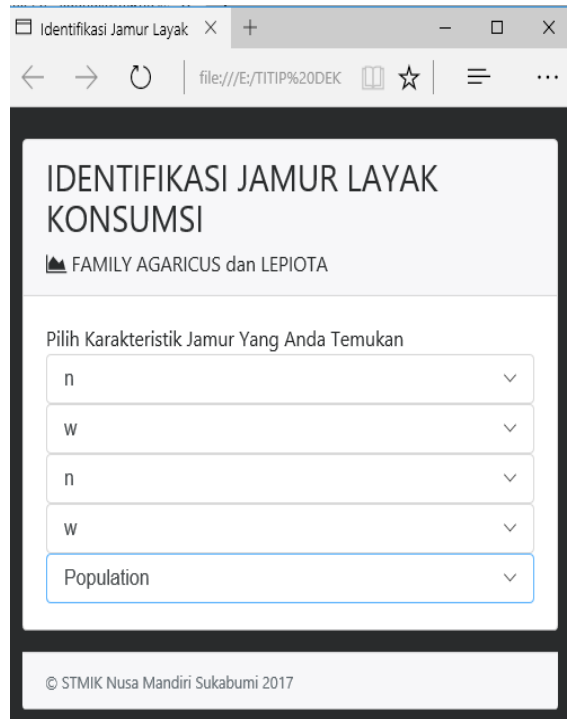


Gambar 3. Alert window identifikasi jamur layak konsumsi



Gambar 4. Alert window identifikasi jamur beracun

Aplikasi akan menampilkan seluruh opsi identifikasi (dari *odor* > *spore print color* > *gill size* > *gill spacing* > *population*) apabila pada opsi-opsi identifikasi sebelumnya, sistem belum dapat menentukan apakah jamur tersebut layak dikonsumsi atau tidak, contoh tampilan seluruh opsi identifikasi dapat dilihat pada gambar 5.



Gambar 5. Contoh opsi identifikasi jamur

IV. PENUTUP

Berdasarkan temuan hasil pengujian galat data menggunakan *10-fold cross validation* bahwa algoritma *decision tree* dengan ketepatan akurasi sebesar 100% dapat digunakan untuk mengidentifikasi jamur *family Agaricus* dan *Lepiota* manakah yang layak dikonsumsi atau tidak, algoritma atau sumber kode program berbasis web dapat dimodifikasi dan diterapkan pada media/bahasa pemrograman lain seperti PC atau perangkat bergerak.

UCAPAN TERIMA KASIH

Terima kasih saya sampaikan kepada kepada editor dan *reviewer* yang telah meluangkan waktu untuk memeriksa dan memberikan masukan dan koreksi pada paper ini.

REFERENSI

- [1] Nandang Suharna, "Keberadaan Basidiomycetes di Cagar Alam Batimurung, Karaenta dan Sekitarnya Maros, Sulawesi Selatan," in *Prosiding Seminar Hasil Litbang SDH*, 1993.
- [2] Wikipedia. Wikipedia Jamur. [Online]. id.wikipedia.org/wiki/jamur
- [3] Meity Suradji Sinaga, *Jamur Merang dan Budidayanya*. Jakarta, Indonesia: Penebar, 2005.
- [4] Mukhamad Khaul Yuhri, "Keanekaragaman Jenis dan Komposisi Jamur Makroskopis di Kawasan Cagar Alam Hutan Gebongan Kecamatan Bergas Kabupaten Semarang," IKIP PGRI Semarang, Semarang, Skripsi 2013.
- [5] Nenda Yunida, Syamswina Syamswina, and Laila Fitri Yeni, "INVENTARISASI JAMUR DI GUNUNG SENUJUJUH KABUPATEN SAMBAS DAN IMPLEMENTASINYA DALAM PEMBUATAN FLASH CARD," *Jurnal Pendidikan dan Pembelajaran*, vol. 3, no. 10, Oktober 2014.
- [6] Achmad Mugiono, Arianthi Tias, and Chotimatul Azmi, *Panduan Lengkap Jamur*. Jakarta, Indonesia: Penebar Swadaya, 2011.
- [7] Bayu Mahardika Putra, "Klasifikasi Jamur ke Dalam Kelas Dapat Dikonsumsi Atau Beracun Menggunakan Algoritma VFI 5 (Studi Kasus : Famili Agaricus dan Lepiota)," IPB, Bogor, Laporan Akhir 2008.
- [8] Galieh Adi and Surya Pradana, "Identifikasi jamur beracun pada jenis jamur famili agaricus dan lepiota berdasarkan klasifikasi," Univeritas Nusantara PGRI Kediri, Kediri, Laporan Akhir 2016.
- [9] Jitbit. (2017, Juni) Helpdesk Software. [Online]. www.jitbit.com/helpdesk/why-web-based
- [10] Wikipedia. Wikipedia - Programming languages used in most popular website. [Online]. en.wikipedia.org/wiki/Programming_languages_used_in_most_popular_websites
- [11] Muhammad Agung Rizkiyana and R. Sandhika Galih Amalga, "Rancangan Arsitektur Aplikasi Pengumpulan Tugas dengan Push Notification Real-Time Menggunakan Node.Js," in *Seminar Nasional Informatika (SEMNASIF)*, Yogyakarta, 2014, pp. 70-75.
- [12] Bambang Setiawan, Raden Sjarief Widjaja, and Setyo Nugroho, "Perancangan Sistem Pendukung Keputusan (SPK) untuk Menentukan Kelaiklautan Kapal," in *Prosiding Seminar Nasional Manajemen Teknologi X*, Surabaya, 2009, pp. C-14-1 – C-14-7.
- [13] Neeraj Bhargava, Girja Sharma, Ritu Bhargava, and Manish Mathuria, "Decision Tree Analysis on J48 Algorithm for Data Mining," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 6, pp. 1114-1119, June 2013.
- [14] Anonymous. numhack. [Online]. <http://numhack.com/decision-tree-learning-avoiding-overfitting/>
- [15] Ian H. Whitten, Eibe Frank, Mark A Hall, and Christopher J Pal, *Data Mining Practical Machine Learning Tools and Techniques*, 4th ed. Massachusetts, United State: Elsevier, 2017.