Identifikasi Jamur yang dapat Dimakan Menggunakan Metode Klasifikasi SVM Berdasarkan Ciri Tekstur

Fikri Azhari1 , Brahma Aditama2 , Adrian Dwi Adinata3 , Adithya Kharisma4

Program Studi Informatika Fakultas Ilmu Komputer

Universitas Pembangunan Nasional Veteran Jakarta

Jl. RS. Fatmawati, Pondok Labu, Jakarta Selatan, DKI Jakarta, 12460, Indonesia.

Fikri Azhari@email1 , Brahma Aditama@email2 , Adrian Dwi Adinata@email3 , adithyak@upnvj.ac.id4

**Abstrak** - Varietas jamur sangatlah banyak, diantara varietas jamur tersebut ada yang beracun dan ada yang dapat dimakan. Akibatnya masyarakat yang kurang pengetahuannya mengenai jamur menjadi takut untuk memakan jamur ketika berada di alam bebas, oleh karena itu penelitian ini memanfaatkan teknologi dalam bidang informatika berupa citra untuk membedakan jamur yang beracun dan yang dapat dimakan menggunakan bahasa pemrograman python dari gabungan ilmu pengolahan citra digital yaitu menggunakan *Low Binary Pattern* dan data mining menggunakan klasifikasi SVM (*Support Vector Machine*).

***Kata Kunci*:** Citra, Python, *Support Vector Machine* (SVM), *Low Binary Pattern*, Jamur

1. **PENDAHULUAN**

Jamur memiliki varietas yang banyak di alam bebas, Diperkirakan terdapat 1,5 juta spesies jamur di dunia. Diantara varietas tersebut ada jamur yang mengandung racun yang berbahaya jika dikonsumsi oleh manusia dan ada jamur yang tidak berbahaya jika dikonsumsi karena tidak memiliki racun, bahkan ada diantara spesies jamur yang bisa langsung dimakan tanpa dimasak terlebih dahulu.

Untuk mengurangi rasa cemas ketika memakan jamur di alam bebas karena kurangnya pengetahuan mengenai varietas jamur, maka varietas jamur harus diklasifikasikan sesuai dengan adanya racun yang berbahaya atau tidak yang terdapat pada jamur.

Pada awalnya klasifikasi jamur dilakukan secara manual dengan mengamati tekstur, bentuk dan warna yang terdapat pada jamur secara langsung. Metode klasifikasi secara manual dapat memakan banyak waktu karena banyaknya kemiripan diantara varietas jamur, oleh karena itu dibutuhkan pengklasifikasian jamur secara cepat.

Pengklasifikasian jamur dapat dilakukan secara cepat dengan memanfaatkan ilmu di bidang informatika yaitu dengan mengenali ciri jamur dari pengolahan citra digital berupa citra lalu diklasifikasikan menggunakan algoritma data mining yaitu *Support Vector Machine*(SVM).

Algoritma *Support Vector Machine* (SVM) adalah salah satu metode dalam supervised learning yang biasanya digunakan untuk klasifikasi dan regresi. SVM merupakan sebuah teknik yang menggunakan 2 titik (2 vektor), yang selanjutnnya 2 titik ini akan membentuk garis pembatas atau sisi pembatas jika 3 dimensi atau lebih. Garis/sisi pembatas yang dibentuk dari 2 buah vektor ini disebut dengan *hyperplane*.

Terdapat banyak penelitian terdahulu yang bersangkutan dengan dilakukannya penelitian ini seperti penelitian Theresia Aruan, 2017. Penelitian ini memiliki tujuan untuk melakukan identifikasi terhadap jamur beracun. Pengolahan citra dilakukan dengan segmentasi dan deteksi tepi lalu melakukan ekstrasi ciri setelah itu diklasifikasikan menggunakan *K-Nearest Neighbour*. Data citra jamur yang digunakan berjumlah 40 citra dengan 8 jenis jamur dan masing-masing jenis jamur berjumlah 5 citra. Data training berjumlah 40% sementara data testing berjumlah 60%. Akurasi yang didapatkan dari klasifikasi menggunakan *K-Nearest Neighbour* adalah 90%.

1. **TINJAUAN PUSTAKA**

**Jamur**

Fungi atau jamur merupakan tumbuhan yang tidak mempunyai klorofil sehingga bersifat heterotrof, tipe sel eukarotik. Jamur ada yang uniseluler dan multiseluler. Tubuhnya terdiri dari benang-benang yang disebut hifa yang dapat membentuk anyaman bercabang-cabang (miselium). [2].

**Citra Digital**

Citra merupakan representasi dua dimensi dari dunia visual. Citra dapat berupa gambar dua dimensi seperti foto dan lukisan maupun tiga dimensi seperti patung. Pada penelitian ini penulis menggunakan citra digital. Citra yang dapat diproses oleh computer merupakan citra digital, suatu citra digital dapat diartikan juga sebagai sebuah matriks yang terdiri dari M kolom dan N baris, dimana perpotongan antara kolom dengan baris disebut pixel [3].

**Pengolahan Citra**

Pengolahan Citra merupakan teknik perbaikan citra dari berbagai gangguan (*Noise*) sehingga mudah diinterpretasikan oleh manusia maupun mesin, citra tersebut dimanipulasi sehingga menghasilkan kualitas yang lebih baik [1].

**Citra Grayscale**

Citra grayscale adalah citra digital yang hanya mempunyai satu nilai kanal pada setiap pikselnya, dengan kata lain nilai bagian red = green = blue. Citra grayscale memiliki kedalaman warna 8 bit (256 kombinasi warna abu-abu [2].

**Ekstraksi Ciri**

Ekstraksi ciri merupakan proses untuk memperoleh ciri objek dalam citra. Proses pengambilan ciri ini memungkinkan pendeteksian tepi, menghitung komponen objek dalam citra, membentuk citra biner hasil perubahan citra masukkan, melakukan penambahan dan pengurangan pola, dan lain sebagainya [1].

**Data Mining**

Data Mining adalah suatu tindakan mengekstrak informasi dan pengetahuan berharga dari suatu data set berukuran besar dengan memanfaatkan teknik tertentu. Informasi yang diperoleh dari proses menambang data ini dapat dipakai untuk pengambilan keputusan [3].

***Support Vector Machine* (SVM)**

*Support Vector Machine* (SVM) adalah suatu teknik untuk melakukan prediksi, baik dalam kasus klasifikasi maupun regresi (Santosa, 2007). SVM memiliki prinsip dasar linier dapat dipisahkan, namun SVM telah dikembangkan agar dapat bekerja pada problem non-linear dengan memasukkan konsep kernel pada ruang kerja berdimensi tinggi. Pada ruang dimensi tinggi, akan dicari *hyperplane* yang dapat memaksimalkan jarak antara kelas data. [4]

1. **METODOLOGI PENELITIAN**

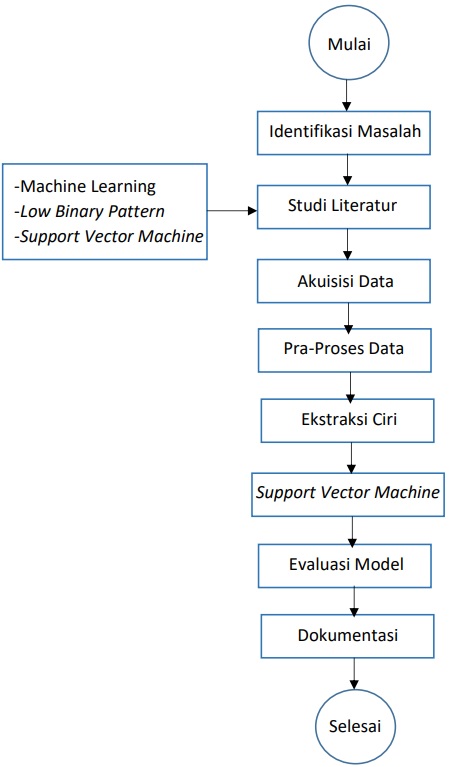
**Identifikasi Masalah**

Identifikasi masalah dalam penelitian ini lebih fokus dalam menjelaskan penerapan ekstraksi ciri menggunakan *Local Binary Pattern* (LBP) dan fitur pada citra jamur lalu melakukan klasifikasi pada citra jamur untuk mengetahui jenis jamur beracun menggunakan algoritma klasifikasi *Support Vector Machine* (SVM).

**Studi Literatur**

Studi literatur digunakan oleh peneliti untuk mengumpulkan berbagai macam data dan informasi yang diperlukan dalam sebuah penelitian. Data dan informasi didapat dari jurnal dan sumber lainnya.

**Kerangka Pikir**

**  
Gambar 1. Kerangka Pikir**

**Akuisisi Data**

Akuisisi data diperlukan untuk mendapatkan data penelitian, data yang dimaksud dalam penelitian ini adalah data citra jamur yang akan digunakan untuk data latih dan data uji. Dalam penelitian ini, proses pengumpulan data yang digunakan mencakup:

1. Observasi, di dalam penelitian ini, penulis akan melaksanakan observasi secara daring.
2. Dokumentasi, dalam penelitian ini, penulis mengambil beberapa dokumentasi citra jamur dengan berbagai macam jenis yang tersedia secara daring.

**Pra-Proses**

Pra proses digunakan pada data citra jamur sebelum diproses dengan metode LBP dan

klasifikasi SVM, yaitu dengan cara resize citra menjadi ukuran 200 x 200 pixel agar setiap ukuran jamur sama rata dan berukuran kecil sehingga memudahkan proses pengolahan citra pada proses berikutnya.

**Ekstraksi Ciri**

Ekstraksi ciri dalam citra jamur menggunakan *Low Binary Pattern* (LBP). Jadi setelah pra proses dilakukan, kemudian citra diubah menjadi *grayscale* setelah itu ciri citra jamur diekstrak menggunakan *Low Binary Pattern* (LBP) lalu data citra dibagi menjadi data latih yang berjumlah 70% dan data testing berjumlah 30%.

**Klasifikasi Support Vector Machine**

Selanjutnya SVM digunakan untuk proses klasifikasi data citra. SVM bertugas untuk menentukan garis pembatas (*hyperplane*) terbaik, yang bisa memberikan jarak terjauh antara 2 kelas yang berbeda.

**Evaluasi Model**

Setelah citra selesai di klasifikasi menggunakan *Supoort Vector Machine* (SVM) dapat diketahui tingkat akurasi pengujian SVM dengan cara mengukur hasil keluarannya.

**Dokumentasi**

Setelah semua tahapan selesai dilakukan, maka tahap terakhir adalah melakukan dokumentasi yang dimana semua hal yang dilakukan terkait dengan penelitian ini ditulis untuk dijadikan laporan penulisan yang disusun secara sistematis.

1. **HASIL DAN PEMBAHASAN**

**Akuisisi Data**

Dalam penelitian ini digunakan sebanyak 48 buah data citra jamur dari 6 ordo jamur yang berbeda, yang dibagi ke dalam dua kelas, yaitu jamur yang bisa dimakan dan jamur yang beracun, masing-masing sebanyak 24 buah data citra.

**Pra Proses**

Pra proses dilakukan setelah akuisisi citra berhasil diselesaikan. data citra yang sudah didapat lalu di pra - proses dengan cara dibaca terlebih dahulu sebagai citra grayscale dengan fungsi opencv imread(), lalu di-resize ukurannya menjadi 200 x 200 pixel dengan menggunakan fungsi dari library opencv yaitu cv2.resize(), lalu diberi label. Resize ini dilakukan agar semua data citra menjadi berukuran sama dan agar data yang diproses ini ukurannya tidak terlalu besar sehingga proses-proses kedepannya dapat dilakukan dengan waktu yang lebih singkat. Setelah itu data nya di dimasukkan kedalam satu list dan setelah itu list ini disimpan sebagai dataset .pickle dengan fungsi dari library pickle, ini dilakukan agar dataset bisa langsung dibaca tanpa perlu melakukan pra-proses setiap kali program dijalankan.

**Ekstraksi Ciri**

Setelah pra-proses selesai dilakukan, tahap selanjutnya adalah ekstraksi ciri. Metode ekstraksi ciri yang digunakan adalah *Gray Pixel Value as Feature* dan *Local Binary Pattern (LBP)*.

*Gray Pixel Value as Feature* mengambil setiap nilai piksel abu-abu pada data citra sebagai array, lalu dengan fungsi .flatten() dari numpy, array nya diubah menjadi array 1 dimensi.

*Local Binary Pattern* adalah sebuah operator yang dapat mengekstrak ciri tekstur dari sebuah gambar dengan cara melabeli piksel gambar dengan membatasi lingkungan setiap piksel dan menganggap hasilnya sebagai bilangan biner.

Ekstrasi LBP memberikan hasil sebagai berikut:

Ciri *Grayscale Pixel Values as Feature* untuk data train adalah:

[array([82, 86, 87, ..., 42, 61, 63], dtype=uint8), array([33, 15, 53, ..., 89, 81, 73], dtype=uint8), array([120, 106, 124, ..., 21, 22, 12], dtype=uint8), array([150, 158, 163, ..., 79, 95, 101], dtype=uint8), array([104, 87, 94, ..., 28, 124, 138], dtype=uint8), array([180, 137, 63, ..., 29, 29, 21], dtype=uint8), array([68, 36, 13, ..., 53, 58, 37], dtype=uint8), array([ 7, 7, 7, ..., 145, 140, 136], dtype=uint8), array([ 30, 29, 30, ..., 127, 132, 141], dtype=uint8), array([156, 171, 164, ..., 60, 32, 35], dtype=uint8), array([ 81, 79, 77, ..., 129, 104, 76], dtype=uint8), array([61, 84, 57, ..., 16, 22, 10], dtype=uint8), array([ 64, 58, 60, ..., 131, 101, 91], dtype=uint8), array([166, 176, 139, ..., 106, 155, 220], dtype=uint8), array([143, 156, 143, ..., 50, 62, 108], dtype=uint8), array([ 78, 61, 76, ..., 96, 90, 107], dtype=uint8), array([46, 59, 53, ..., 58, 55, 47], dtype=uint8), array([150, 158, 163, ..., 79, 95, 101], dtype=uint8), array([13, 13, 12, ..., 31, 31, 30], dtype=uint8), array([47, 21, 21, ..., 54, 73, 81], dtype=uint8), array([ 75, 76, 77, ..., 58, 71, 106], dtype=uint8), array([4, 5, 6, ..., 1, 1, 6], dtype=uint8), array([27, 21, 8, ..., 37, 31, 39], dtype=uint8), array([116, 113, 111, ..., 45, 40, 47], dtype=uint8), array([ 30, 28, 25, ..., 77, 138, 16], dtype=uint8), array([107, 96, 95, ..., 247, 174, 212], dtype=uint8), array([173, 176, 173, ..., 112, 51, 73], dtype=uint8), array([ 84, 73, 23, ..., 34, 110, 72], dtype=uint8), array([128, 122, 119, ..., 83, 102, 25], dtype=uint8), array([ 29, 30, 30, ..., 124, 123, 122], dtype=uint8), array([68, 68, 75, ..., 61, 66, 65], dtype=uint8), array([13, 13, 12, ..., 31, 31, 30], dtype=uint8), array([40, 40, 39, ..., 72, 75, 83], dtype=uint8), array([ 51, 52, 50, ..., 212, 201, 251], dtype=uint8)]

Sedangkan ciri *Grayscale Pixel Value as Feature* untuk data test adalah:

[array([140, 146, 148, ..., 126, 91, 124], dtype=uint8), array([ 65, 67, 95, ..., 162, 164, 165], dtype=uint8), array([126, 118, 110, ..., 41, 27, 18], dtype=uint8), array([154, 148, 132, ..., 87, 38, 114], dtype=uint8), array([22, 26, 40, ..., 94, 97, 54], dtype=uint8), array([106, 117, 144, ..., 68, 54, 33], dtype=uint8), array([230, 162, 118, ..., 15, 80, 119], dtype=uint8), array([160, 141, 146, ..., 130, 143, 134], dtype=uint8), array([ 89, 97, 102, ..., 31, 43, 95], dtype=uint8), array([ 96, 106, 136, ..., 94, 107, 86], dtype=uint8), array([136, 137, 130, ..., 18, 37, 13], dtype=uint8), array([ 80, 131, 131, ..., 167, 141, 151], dtype=uint8), array([ 2, 43, 39, ..., 48, 31, 3], dtype=uint8), array([68, 70, 61, ..., 18, 13, 52], dtype=uint8), array([57, 57, 58, ..., 44, 45, 44], dtype=uint8)]

Ciri LBP untuk data training (34 data citra):

[array([239, 95, 0, ..., 138, 9, 0], dtype=uint8), array([ 26, 57, 32, ..., 1, 135, 131], dtype=uint8), array([255, 247, 255, ..., 15, 135, 131], dtype=uint8), array([ 7, 255, 7, ..., 131, 143, 0], dtype=uint8), array([ 40, 216, 4, ..., 12, 4, 130], dtype=uint8), array([ 1, 199, 135, ..., 0, 143, 2], dtype=uint8), array([ 55, 255, 127, ..., 15, 13, 2], dtype=uint8), array([ 32, 224, 216, ..., 8, 0, 131], dtype=uint8), array([255, 127, 127, ..., 10, 1, 131], dtype=uint8), array([187, 32, 192, ..., 128, 132, 130], dtype=uint8), array([255, 127, 255, ..., 135, 135, 131], dtype=uint8), array([253, 62, 63, ..., 141, 14, 2], dtype=uint8), array([ 29, 8, 128, ..., 131, 143, 1], dtype=uint8), array([ 0, 192, 255, ..., 139, 129, 128], dtype=uint8), array([ 30, 7, 231, ..., 135, 143, 1], dtype=uint8), array([223, 54, 255, ..., 131, 129, 131], dtype=uint8), array([ 8, 8, 8, ..., 141, 12, 2], dtype=uint8), array([231, 255, 255, ..., 141, 8, 0], dtype=uint8), array([ 0, 137, 0, ..., 143, 4, 131], dtype=uint8), array([ 4, 255, 9, ..., 143, 3, 131], dtype=uint8), array([248, 48, 240, ..., 133, 139, 1], dtype=uint8), array([193, 252, 126, ..., 3, 131, 131], dtype=uint8), array([ 56, 32, 224, ..., 8, 0, 130], dtype=uint8), array([ 0, 128, 128, ..., 131, 131, 131], dtype=uint8), array([ 63, 127, 127, ..., 143, 143, 3], dtype=uint8), array([ 61, 98, 193, ..., 0, 140, 0], dtype=uint8), array([ 20, 182, 255, ..., 142, 3, 131], dtype=uint8), array([255, 255, 239, ..., 129, 128, 129], dtype=uint8), array([249, 8, 0, ..., 137, 12, 0], dtype=uint8), array([ 56, 32, 248, ..., 129, 141, 0], dtype=uint8), array([255, 247, 255, ..., 15, 135, 131], dtype=uint8), array([188, 98, 223, ..., 139, 1, 131], dtype=uint8), array([193, 254, 25, ..., 12, 10, 0], dtype=uint8), array([254, 251, 117, ..., 143, 6, 131], dtype=uint8)]

Sedangkan ciri LBP untuk data test (14 data citra) adalah:

[array([ 0, 192, 207, ..., 143, 10, 1], dtype=uint8), array([248, 96, 192, ..., 143, 15, 3], dtype=uint8), array([ 63, 247, 247, ..., 8, 8, 0], dtype=uint8), array([ 16, 164, 250, ..., 15, 3, 131], dtype=uint8), array([199, 199, 255, ..., 3, 139, 0], dtype=uint8), array([ 54, 247, 247, ..., 133, 143, 2], dtype=uint8), array([227, 219, 56, ..., 0, 130, 131], dtype=uint8), array([ 8, 8, 8, ..., 141, 12, 2], dtype=uint8), array([ 0, 226, 255, ..., 8, 0, 131], dtype=uint8), array([ 8, 8, 16, ..., 143, 15, 3], dtype=uint8), array([248, 104, 0, ..., 8, 0, 128], dtype=uint8), array([ 56, 120, 112, ..., 4, 143, 3], dtype=uint8), array([192, 176, 248, ..., 128, 128, 128], dtype=uint8), array([112, 240, 224, ..., 7, 143, 3], dtype=uint8), array([233, 64, 240, ..., 136, 8, 0], dtype=uint8)]

**Hasil Klasifikasi SVM**

Setelah ciri sudah berhasil di ekstrak, hasilnya dibuat menjadi xtran dan xtest, dengan ytrain dan ytest sebagai labelnya, variabel ini lalu digunakan untuk melatih model klasifikasi SVM, dengan setting parameter C(Regularization) = 1, kernel = ‘poly’, dan gamma = ‘auto’.

Dari 5 kali percobaan, akurasi yang didapat adalah seperti berikut:

|  |  |
| --- | --- |
| ***Grayscale Value as Feature*** | |
| 1 | 93% |
| 2 | 100% |
| 3 | 86% |
| 4 | 80% |
| 5 | 100% |
| Rata-rata= 92.8% | |

|  |  |
| --- | --- |
| **Local Binary Pattern (LBP)** | |
| 1 | 86% |
| 2 | 80% |
| 3 | 73% |
| 4 | 80% |
| 5 | 80% |
| Rata-rata= 79.8% | |

Dari percobaan diatas didapat bahwa klasifikasi dengan menggunakan nilai piksel abu-abu ternyata memberikan hasil yang lebih akurat daripada ekstraksi ciri dengan LBP dengan perbedaan rata-rata akurasi sebesar kurang lebih 10%.

1. **KESIMPULAN DAN SARAN**

**Kesimpulan**

Untuk pengklasifikasian jamur, metode ekstraksi ciri *Local Binary Pattern (LBP)* dinilai kurang tepat, karena walaupun dapat menghasilkan akurasi sebesar 79.8%, masih kalah dengan ekstraksi ciri nilai piksel abu-abu yang menghasilkan akurasi sebesar 92.8%. Hal ini mungkin disebabkan karena *background* citra jamur yang kompleks.

**Saran**

1. Memperbanyak jumlah citra yang digunakan.
2. Menggunakan metode pra-proses yang lebih beragam.
3. **DAFTAR PUSTAKA**

[1] Theresia Aruan. 2017. Identifikasi Jenis Tanaman Jamur Beracun Menggunakan Pendekatan *K-Nearest Neighbour*, Medan: Universitas Sumatera Utara.

[2] Arma, Rinci dan Hendro. 2020. Penerapan Naive Bayes Gaussian pada Klasifikasi Jenis Jamur Berdasarkan Ciri Statistik Orde Pertama. Surabaya: Institut Teknologi Adhi Tama Surabaya.

[3] Duwen, Ermatita, Noor Falih. 2021. Penggunaan *K-Nearest Neighbour* untuk Mengklasifikasi Citra Belimbing Berdasarkan Fitur Warna. Jakarta: Universitas Pembangunan Nasional “Veteran” Jakarta.

[4] Pusphita, Yuciana, Dwi. 2014. Penerapan Metode Klasifikasi *Support Vector Machine* (SVM) pada Data Akreditasi Sekolah Dasar (SD) di Kabupaten Magelang. Semarang: Universitas Diponegoro.