**Naskah Publikasi**

**Klasifikasi Jenis Buah Anggur Menggunakan Metode Ekstraksi Fitur Tekstur Dan Jaringan Saraf Tiruan**

PROYEK TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat

Mencapai derajat sarjana S-1

Program Studi Teknik Informatika



Disusun Oleh

**Heri Andria Herdiana**

**3125111378**

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS BISNIS DAN TEKNOLOGI INFORMASI

UNIVERSITAS TEKNOLOGI YOGYAKARTA

2017

Naskah Publikasi

**Klasifikasi Jenis Buah Anggur Menggunakan Metode Ekstraksi Fitur Tekstur Dan Jaringan Saraf Tiruan**

Disusun oleh

Heri Andria Herdiana

3125111378

Telah Disetujui oleh Pembimbing

Pembimbing

Suhirman, Ph.D Tanggal : ........................................

Klasifikasi Jenis Buah Anggur Menggunakan Metode

Ekstraksi Fitur Tekstur Dan Jaringan Saraf Tiruan

Heri Andria Herdiana

*Program Studi Teknik Informatika,Fakultas Bisnis dan Teknologi Informasi  
Universitas Teknologi Yogykarta*

*Jl. Ringroad Utara Jombor Sleman Yogyakarta*

*E-mail :* [*email.mahasiswa@gmail.com*](mailto:email.mahasiswa@gmail.com)

## ABSTRAK

*Anggur merupakan tanaman buah berupa perdu merambat yang termasuk ke dalam keluarga Vitaceae. Berbagai jenis buang anggur bisa kita jumpai dengan mudah di pasar serba ada ataupun di toko buah pinggir jalan sekalipun. Terkadang mereka kesulitan dalam memilah satu jenis anggur dengan jenis anggur yang lain. Dengan berkembang pesatnya teknologi komputer, seperti halnya manusia pada saat ini komputer bisa melakukan banyak hal yang sebelumnya hanya bisa dilakukan manusia. Salah satunya adalah dalam proses pengklasifikasian atau pemilahan benda atau objek. Dengan memanfaatkan teknik pemrosesan citra, komputer dapat digunakan untuk melakukan pengklasifikasian objek atau benda. Teknik pengklasifikasian tersebut akan menggunakan metode ekstraksi fitur dari citra buah dan jaringan saraf tiruan untuk proses klasifikasi jenis buah anggur. Proses klasifikasi diawali dengan membuat matriks GLCM dari citra diikuti dengan proses ekstraksi properti GLCM, kemudian properti tersebut digunakan sebagai nilai masukan untuk jaringan saraf backpropagation. Hasil keluaran dari sistem pengklasifikasi anggur ini berupa informasi mengenai deskripsi dan juga jenis buah anggur dari citra masukan yang bersangkutan. Akurasi dari hasil keluaran sistem mencapai 100% untuk data citra terlatih, 91.67% untuk data citra tes atau tidak terlatih yang dicapai dengan menggunakan 16 hidden neuron..*

Kata kunci : Adeline, Citra, Pemrosesan, Otomasi

### 1. PENDAHULUAN

Anggur merupakan tanaman buah berupa perdu merambat yang termasuk ke dalam keluarga Vitaceae. mudah di pasar serba ada ataupun di toko buah pinggir jalan sekalipun. Selain itu buah anggur juga banyak diminati di Indonesia, tetapi tidak semua orang mengetahui jenis-jenis buah anggur. Terkadang mereka kesulitan dalam memilah satu jenis anggur dengan jenis anggur yang lain. Tak jarang sering terjadi kesalahan dalam memilih jenis anggur yang sesuai. Namun dengan berkembang pesatnya teknologi komputer, saat ini komputer dapat melakukan banyak hal yang sebelumnya hanya bisa dilakukan manusia. Salah satunya adalah dalam proses pengklasifikasian atau pemilahan benda atau objek. Seperti halnya ditunjukkan pada penelitian yang dilakukan oleh [1], mengenai pengklasifikasian jenis buah apel dengan menggunakan algoritma *k-nearest neighbor* (K-NN) dan ekstraksi fitur. Dari hasil penelitian tersebut menunjukkan bahwa metode yang digunakan dapat mengenali jenis buah apel dengan cukup baik, yaitu dengan tingkat akurasi mencapai 90%. Begitu pula penelitian yang dilakukan oleh [2], mengenai identifikasi jenis tanaman buah tropika berdasarkan citra daun. Identifikasi dilakukan dengan metode ekstraksi fitur *gray level co-occurrence matrix* (GLCM) dari tekstur citra permukaan daun buah tropika yang kemudian digunakan sebagai masukan dari pelatihan jaringan saraf tiruan untuk proses identifikasi. Secara keseluruhan pada penelitian tersebut menghasilkan hasil akurasi sebesar 90%. Dengan memanfaatkan metode yang terdapat dalam disiplin ilmu *computer vision*, yang salah satunya adalah metode ekstraksi fitur tekstur *gray level co-occurrence matrix* (GLCM). Menjadikan komputer dapat digunakan untuk melakukan proses pengklasifikasian jenis buah anggur. Dengan alasan itulah, penulis mengajukan penelitian lebih lanjut mengenai teknik pengklasifikasian tersebut.

Berdasarkan latar belang tersebut, peneliti membatasi masalah yang diangkat dengan beberapa batasan berikut.

1. Pengklasifikasian jenis buah hanya diperuntukkan buah anggur. Jenis buah anggur yang akan dilakukan pengklasifikasian hanya tiga jenis.
2. Kualitas citra yang ditangkap sistem dipengaruhi oleh kondisi lingkungan sekitar seperti tingkat pencahayaan yang rendah serta kualitas kamera pada perangkat.
3. Hasil atau keluaran dari sistem adalah berupa penggolongan nama jenis buah anggur dengan menggunakan metode ekstraksi fitur tekstur *Gray Level Co-Occurrence Matrix* (GLCM ) dari citra buah dan jaringan saraf tiruan *backpropagation* untuk proses klasifikasi jenis buah anggur.

Tujuan dari penelitian ini adalah membangun sistem klasifikasi jenis buah anggur menggunakan metode ekstraksi fitur tekstur GLCM dan jaringan saraf tiruan *backpropagation* yang dapat digunakan untuk membantu proses pemilahan jenis buah anggur.

### 2. LANDASAN TEORI

##### **2.1. Ekstraksi Fitur Tekstur**

Dalam buku Teori dan Aplikasi Pengolahan Citra, [3] mendefinisikan tekstur sebagai hubungan mutual antara nilai intensitas piksel-piksel yang bertetangga yang berulang di suatu area yang lebih luas daripada jarak hubungan tersebut. Berdasarkan keteraturan pengulangan pola dalam objek, tekstur dapat dikategorikan ke dalam dua bentuk, yaitu tekstur teratur dan tekstur tidak teratur. Tekstur teratur cenderung tekstur hasil buatan manusia, sedangkan tekstur alamiah bersifat tidak teratur. Berdasarkan strukturnya, tekstur dapat diklasifikasikan dalam dua golongan, yaitu: mikro tekstur dan makro tekstur. Analisis tekstur dari sebuah citra dapat dimanfaatkan sebagai proses antara untuk melakukan klasifikasi dan interpretasi citra. Metode yang digunakan untuk memperoleh fitur tekstur dapat dibedakan menjadi tiga golongan, antara lain:

1. Metode Statistik

Metode ini menggunakan perhitungan statistik distribusi derajat ke abuan (histogram) dengan mengukur tingkat kekontrasan, granularitas, dan kekasaran suatu daerah dari hubungan ketetanggaan antar piksel di dalam citra. Paradigma statistik ini penggunaannya tidak terbatas, sehingga sesuai untuk tekstur-tekstur alami yang tidak terstruktur dari sub pola dan himpunan aturan (mikro struktur) [4]. Metode ini memiliki dua jenis orde, yaitu ekstraksi ciri orde pertama yang dilakukan melalui histogram citra, dan ekstraksi ciri statistik orde kedua dilakukan dengan matriks *co-occurrence*, yaitu suatu matriks antara yang merepresentasikan hubungan ketetanggaan antar piksel dalam citra pada berbagai arah orientasi dan jarak spasial.

1. Metode Spektral

Metode ini berdasarkan pada fungsi auto-korelasi suatu daerah atau *power distribution* pada domain transformasi *Fourier* dalam mendeteksi periodisitas tekstur.

1. Metode Struktural

Analisis dengan metode struktural ini menggunakan deskripsi primitif tekstur dan aturan sintaktis. Metode struktural banyak digunakan untuk pola-pola makro struktur, yakni pola yang memiliki perulangan pola lokal secara periodik pada suatu daerah citra, biasanya terdapat pada pola-pola buatan manusia dan cenderung mudah untuk direpresentasikan secara matematis.

##### **2.2. Gray Level Co-Occurrence Matrix (GLCM)**

Menurut [5] dalam bukunya Pengolahan Citra Digital Dan Aplikasinya, *Gray Level Co-Occurrence Matrix* (GLCM) atau matriks intensitas *co-occurrence* adalah suatu matriks yang menggambarkan frekuensi munculnya pasangan dua piksel dengan intensitas tertentu dalam jarak dan arah tertentu dalam citra. Matriks *co-occurrence* sangat diperlukan untuk membantu perhitungan sifat-sifat yang akan diekstrak dari citra tersebut.

Sebuah fitur untuk mengukur keteracakan dari distribusi intensitas disebut entropi dan didefinisikan sebagai berikut:

|  |  |
| --- | --- |
|  | (1) |

Nilai entropi maksimum jika semua elemen p(i,j) sama, yaitu matriks yang berhubungan dengan citra di mana tidak terdapat susunan tertentu dalam pasangan intensitas dengan jarak vektor tertentu (daerah konstan).

Fitur yang lain adalah energi, yaitu fitur untuk mengukur konsentrasi pasangan intensitas pada matriks *co-occurrence*, dan didefinisikan dengan:

|  |  |
| --- | --- |
|  | (2) |

Nilai energi makin membesar jika pasangan piksel yang memenuhi syarat matriks intensitas *co-occurrence* terkonsentrasi pada beberapa koordinat dan mengecil jika letaknya menyebar. Fitur lainnya yaitu kontras yang digunakan untuk mengukur kekuatan perbedaan intensitas dalam citra dan dinyatakan dengan:

|  |  |
| --- | --- |
|  | (2 (3) |

Nilai kontras membesar jika variasi intensitas dalam citra tinggi, dan menurun jika variasinya rendah. Kebalikan dari kontras adalah homogenitas, yaitu untuk mengukur kehomogenan variasi intensitas dalam citra, dan didefinisikan dengan:

|  |  |
| --- | --- |
|  | (4) |

Nilai homogenitas membesar jika variasi intensitas dalam citra mengecil dan sebaliknya mengecil jika variasi intensitas dalam citra membesar.

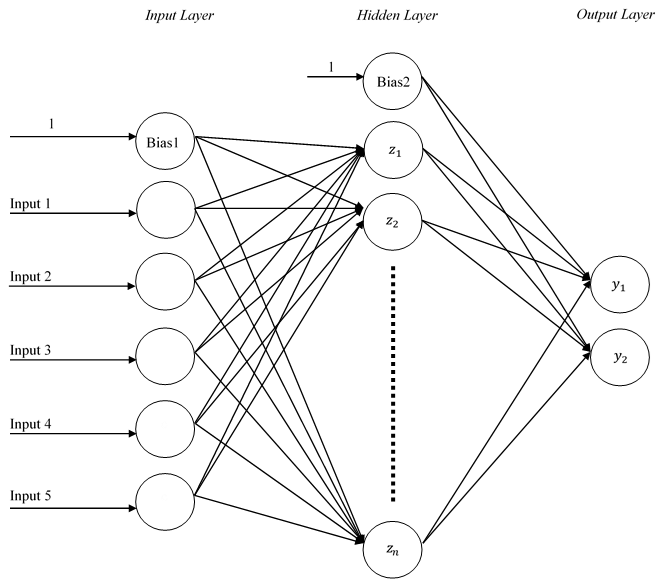
|  |  |
| --- | --- |
|  | ( (5) |

Nilai korelasi menunjukkan ukuran ketergantungan linear derajat ke abuan citra sehingga dapat memberikan petunjuk adanya struktur linear dalam citra.

Dalam lima persamaan di atas, notasi p melambangkan probabilitas yang bernilai mulai dari nol hingga satu, yaitu elemen dalam matriks *co-occurrence*, sedangkan i dan j melambangkan pasangan intensitas yang berdekatan, yang dalam matriks *co-occurrence* masing-masing menjadi nomor baris dan nomor kolom.

##### **2.3. Jaringan Saraf Tiruan Backpropagation**

Menurut [6] jaringan saraf tiruan *backpropagation* merupakan salah satu algoritma yang sering digunakan dalam menyelesaikan masalah-masalah yang rumit. Pada jaringan diberikan sepasang pola yang terdiri atas pola masukan dan pola yang diinginkan. Ketika suatu pola diberikan kepada jaringan, bobot-bobot diubah untuk memperkecil perbedaan pola keluaran dan pola yang diinginkan. Latihan ini dilakukan berulang-ulang sehingga semua pola yang dikeluarkan jaringan dapat memenuhi pola yang diinginkan. Algoritma pelatihan jaringan saraf perambatan galat mundur terdiri atas dua langkah, yaitu perambatan maju dan perambatan mundur. Langkah perambatan maju dan perambatan mundur ini dilakukan pada jaringan untuk setiap pola yang diberikan selama jaringan mengalami pelatihan. Pada Gambar 1 diperlihatkan rancangan dari jaringan saraf tiruan *backpropagation*.



Gambar 1: Rancangan Jaringan Saraf Backpropagation

### 3. METODOLOGI PENELITIAN

Dalam penelitian ini objek yang menjadi pusat perhatian penelitian adalah buah anggur yang terdiri dari tiga jenis yaitu Anggur Merah *Red Globe*, Anggur Merah *Crimson* dan Anggur Hijau *Calmeria*. Dalam dunia industri *wine* ataupun dalam kehidupan sehari-hari terkadang kita merasa sulit dalam membedakan satu jenis anggur dengan jenis lainnya, sehingga cenderung terjadi kesalahan dalam proses pemilahan. Dalam industri anggur sendiri otomatisasi proses pemilahan jenis buah anggur akan membantu meningkatkan efisiensi baik dari segi waktu maupun biaya operasional..

Tahapan penelitian merupakan serangkaian tahapan yang akan dilalui selama peneliti melakukan pengembangan dari sistem pengklasifikasi jenis buah anggur. Pada penelitian ini tahapan penelitian yang digunakan adalah metode ADI, yang dilanjutkan dengan tahap pengujian sistem. Metode ADI sendiri merupakan singkatan dari Analisis Desain dan Implementasi. Gambar 2 memperlihatkan tahapan proses penelitian menggunakan metode ADI.



Gambar 2: Alur Proses Penelitian

##### **3.1. Analisis**

Analisis data dilakukan untuk memastikan setiap data yang dibutuhkan oleh sistem dapat terpenuhi. Data-data tersebut merupakan data citra buah anggur yang terdiri dari 30 buah citra. Di mana data citra tersebut terdiri dari 10 buah Anggur *Red Globe*, 10 buah anggur *Crimson* dan 10 buah anggur *Calmeria*.

##### **3.2. Desain Sistem**

Desain sistem merupakan tahap di mana setiap alur proses sistem di susun dan dirancang sedemikian rupa sehingga proses yang ada pada sistem yang dibangun menjadi terstruktur dan terorganisir dengan baik sesuai dengan kebutuhan.

1. Perancangan Proses Perancangan alur proses sistem secara keseluruhan dilakukan dengan menggunakan *Case Tools* UML atau *Unified Modeling Language* yang terdiri dari *Use Case Diagram, Activity Diagram* dan *Class Diagram* yang menjadi interpreter dari proses-proses yang ada pada sistem. Secara gari besar proses-proses yang terdapat pada sistem terdiri dari proses masukan citra, pengolahan citra, ekstraksi fitur, pelatihan jaringan saraf *backpropagatio*n serta proses pengujian atau pengetesan. Untuk ekstraksi fitur yang digunakan di dalam penelitian adalah menggunakan metode statistik orde ke dua atau yang lebih dikenal dengan metode *gray level co-occurrence matrix* (GLCM).
2. Perancangan Antarmuka Perancangan antarmuka dilakukan berdasarkan kebutuhan sistem dan pengguna, di mana sistem dibangun dengan menggunakan informasi yang telah di dapatkan akan kebutuhan pengguna mengenai kepraktisan sebuah sistem yang dapat digunakan untuk melakukan klasifikasi secara *mobile* , terintegrasi dan juga mampu memberikan keakuratan yang baik dalam melakukan klasifikasi. *Tools* yang digunakan untuk membuat rancangan antarmuka pada penelitian ini adalah Android Studio yang merupakan salah satu *tools* yang banyak digunakan untuk mengembangkan aplikasi berbasis sistem operasi android.

##### **3.3. Implementasi Sistem**

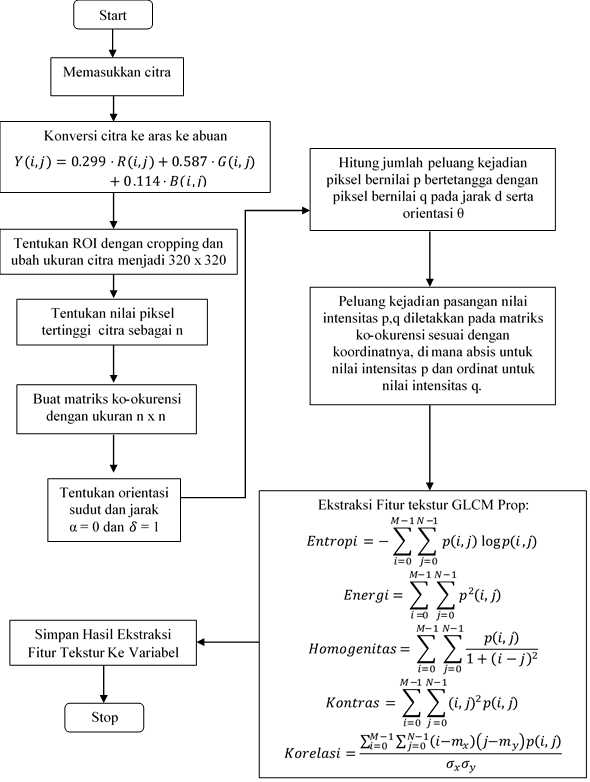
Setelah tahap analisis dan perancangan sistem selesai dilaksanakan, maka tahap selanjutnya adalah melakukan implementasi terhadap rancangan sistem yang telah dibuat pada tahap sebelumnya ke dalam bentuk bahasa pemrograman. Proses-proses yang sudah dirancang akan dikonversikan menjadi serangkaian fungsi atau metode-metode yang ditulis dengan menggunakan bahasa pemrograman Java Android.

##### **3.4. Pengujian Sistem**

Pengujian sistem adalah tahap akhir dalam penelitian ini, di mana pada tahap ini akan dilakukan proses kalkulasi terhadap sistem yang telah dibangun. Kalkulasi yang dilakukan meliputi seberapa baik performa, keakuratan, dan juga kemudahan dalam pengoperasiannya bagi pengguna. Setiap proses inti pada sistem akan di uji apakah sudah memenuhi kriteria yang diharapkan atau tidak.

**4. HASIL DAN PEMBAHASAN**

Proses-proses yang terdapat pada sistem dibagi ke dalam tiga bagian utama yaitu, proses ekstraksi fitur tekstur, proses pelatihan jaringan saraf tiruan dan proses klasifikasi jenis buah anggur.



Gambar 3: Proses Ekstraksi Fitur Tekstur GLCM

Pada Gambar 3 diperlihatkan proses-proses yang terdapat pada ekstraksi fitur tekstur GLCM, Proses ekstraksi fitur tekstur dilakukan dengan menggunakan arah sudut alpha = 0 derajat dan jarak d = 0.



Gambar 4: Proses Pelatihan Jaringan Saraf Tiruan

Pada Gambar 4 diperlihatkan tahapan-tahapan dari proses pelatihan jaringan saraf tiruan yang dilakukan. Proses pelatihan jaringan saraf, terdiri dari umpan maju dan galat mundur. Di mana proses perubahan bobot dilakukan pada tahap galat mundur. Untuk proses klasifikasi jenis buah, tahapan yang dilakukan hampir sama dengan proses pelatihan, hanya saja proses klasifikasi hanya menggunakan umpan maju saja.

Pengujian sistem dilakukan bertujuan untuk mengetahui seberapa baik performa dari sistem yang dibangun. Pada pengujian sistem terdapat dua tahap pengujian yaitu proses pelatihan jaringan dan proses prediksi jenis buah anggur. Proses pelatihan akan dilakukan untuk proses pembelajaran bagi jaringan saraf dalam mengenali citra tekstur buah anggur dan proses prediksi jenis bertujuan untuk mengetahui seberapa akurat sistem dalam mengklasifikasikan citra buah anggur.

Proses pelatihan jaringan dilakukan dengan menggunakan 30 data latih yang terdiri dari 10 data latih untuk buah anggur merah (Red Globe), 10 untuk data latih buah anggur hijau (*Calmeria*) dan 10 data latih untuk buah anggur merah (*Crimson*). Data latih merupakan citra yang diambil dengan menggunakan kamera DSLR Canon EOS 700D, dengan jarak antara lensa kamera dan objek sebesar 25 cm dan tanpa *zoom*. Dari 30 data latih tersebut didapatkan 5 parameter masukan dari setiap masing-masing data latih. Data parameter masukan yang digunakan merupakan hasil ekstraksi fitur tekstur dengan menggunakan metode *gray level co-occurrency matrix* (GLCM) dari masing-masing citra data latih. Pada Tabel 1 ditunjukkan data latih dengan nilai masukan yang sudah dinormalisasi. Proses normalisasi dilakukan dengan cara membagi masing-masing parameter dengan nilai parameter terbesar pada masing-masing kategori parameter.

Tabel 1: Data Latih

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Nama Jenis** | **Entropi** | **Energi** | **Homogenitas** | **Kontras** | **Korelasi** |
| 1 | *Red Globe* | 0.6736 | 0.3002 | 0.9102 | 0.1607 | 0.9998 |
| 2 | *Red Globe* | 0.6023 | 0.8249 | 0.935 | 0.2338 | 0.9991 |
| 3 | *Red Globe* | 0.6227 | 0.855 | 0.897 | 0.2654 | 0.9993 |
| 4 | *Red Globe* | 0.6248 | 0.7422 | 0.9129 | 0.2058 | 0.9995 |
| 5 | *Red Globe* | 0.5607 | 1 | 1 | 0.115 | 0.9993 |
| 6 | *Red Globe* | 0.6102 | 0.9608 | 0.9483 | 0.1879 | 0.9994 |
| 7 | *Red Globe* | 0.6722 | 0.551 | 0.8944 | 0.1899 | 0.9995 |
| 8 | *Red Globe* | 0.6697 | 0.3319 | 0.9185 | 0.2185 | 0.9997 |
| 9 | *Red Globe* | 0.6403 | 0.7653 | 0.8925 | 0.234 | 0.9993 |
| 10 | *Red Globe* | 0.6863 | 0.3024 | 0.8824 | 0.2725 | 0.9996 |
| 11 | *Calmeria* | 0.7456 | 0.4608 | 0.7202 | 0.5622 | 1 |
| 12 | *Calmeria* | 0.7997 | 0.1584 | 0.7132 | 0.5275 | 1 |
| 13 | *Calmeria* | 0.8658 | 0.2379 | 0.6037 | 0.7056 | 0.9996 |
| 14 | *Calmeria* | 0.8211 | 0.1531 | 0.6462 | 0.6567 | 0.9998 |
| 15 | *Calmeria* | 0.7913 | 0.2019 | 0.6788 | 0.6781 | 0.9999 |
| 16 | *Calmeria* | 0.8746 | 0.2154 | 0.5954 | 0.8798 | 0.9996 |
| 17 | *Calmeria* | 0.7906 | 0.2159 | 0.6853 | 0.6353 | 0.9999 |
| 18 | *Calmeria* | 0.775 | 0.369 | 0.6561 | 0.7744 | 0.9998 |
| 19 | *Calmeria* | 0.8924 | 0.0483 | 0.604 | 0.5876 | 0.9999 |
| 20 | *Calmeria* | 0.8302 | 0.0972 | 0.6382 | 0.7055 | 0.9998 |
| 21 | *Crimson* | 0.9737 | 0.0397 | 0.5168 | 0.9162 | 0.9992 |
| 22 | *Crimson* | 0.9567 | 0.0456 | 0.5222 | 1 | 0.9992 |
| 23 | *Crimson* | 0.9792 | 0.0306 | 0.5584 | 0.7645 | 0.9995 |
| 24 | *Crimson* | 1 | 0.0252 | 0.5572 | 0.621 | 0.9995 |
| 25 | *Crimson* | 0.9361 | 0.0618 | 0.5822 | 0.7086 | 0.9996 |
| 26 | *Crimson* | 0.9734 | 0.0456 | 0.5597 | 0.8023 | 0.9996 |
| 27 | *Crimson* | 0.9452 | 0.0548 | 0.5668 | 0.9227 | 0.9995 |
| 28 | *Crimson* | 0.9679 | 0.0231 | 0.514 | 0.8841 | 0.9993 |
| 29 | *Crimson* | 0.9589 | 0.0279 | 0.6025 | 0.592 | 0.9998 |
| 30 | *Crimson* | 0.9643 | 0.0258 | 0.5802 | 0.6978 | 0.9996 |

Pada pelatihan yang penulis lakukan, digunakan metode *adaptive learning rate* pada jaringan saraf dengan konfigurasi jaringan saraf tiruan yang berbeda. Penulis melakukan percobaan terhadap pelatihan dengan menggunakan jumlah *hidden neuron* yang berbeda. Selain jumlah *hidden neuron*, parameter pelatihan jaringan saraf lainnya bernilai sama, data parameter yang digunakan secara rinci dapat dilihat pada Tabel 2 .

Tabel 2: Parameter Pelatihan Jaringan Saraf

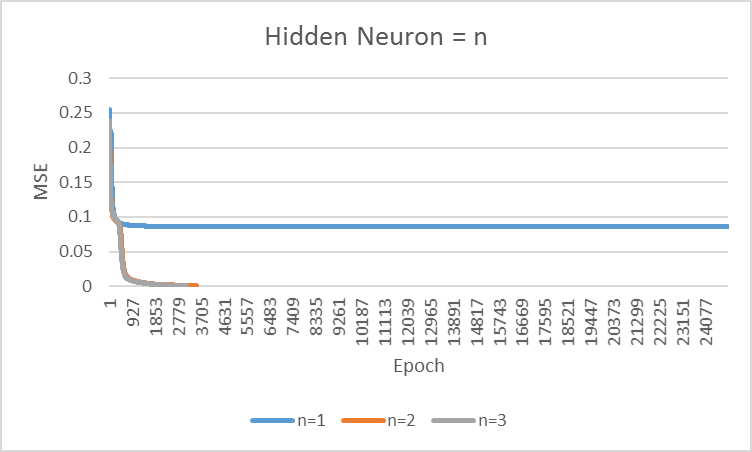
|  |  |
| --- | --- |
| Jumlah sel lapisan masukan | 5 |
| Jumlah sel lapisan keluaran | 2 |
| Galat yang diijinkan | 0.001 |
| *Learning rate* | 0.25 |
| Kenaikan *Learnnig rate* | 1.0001 |
| Penurunan *Learning rate* | 0.9999 |
| Fungsi aktivasi | *logsig* |

Dengan menggunakan nilai-nilai parameter yang tertera pada Tabel 2 dan beberapa konfigurasi jumlah *hidden neuron* yang berbeda, didapatkan hasil pelatihan yang cukup variatif. Hasil dari percobaan pelatihan jaringan dengan menggunakan beberapa konfigurasi jumlah *hidden neuron* yang berbeda dapat dilihat pada Tabel 3.

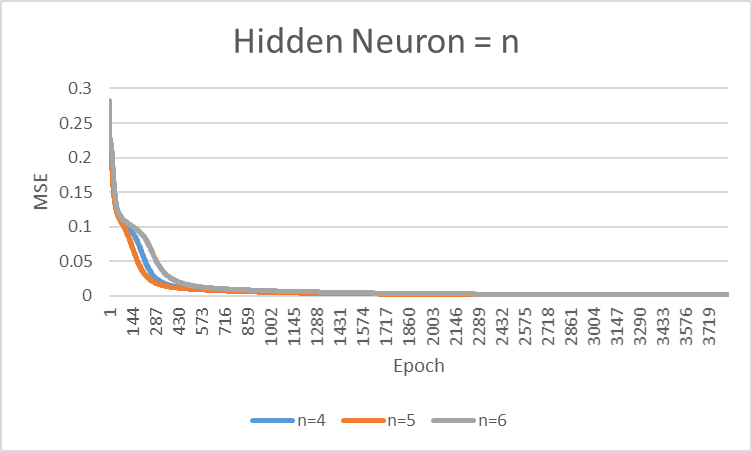
Tabel 3: Konvergensi Hasil Pelatihan

|  |  |  |  |
| --- | --- | --- | --- |
| **Jumlah Hidden Neuron** | **Epochs** | **MSE** | **Status Konvergen** |
| 1 | 25000 | 0.0855 | Tidak |
| 2 | 3514 | 0.001 | Iya |
| 3 | 3135 | 0.001 | Iya |
| 4 | 2963 | 0.001 | Iya |
| 5 | 3507 | 0.001 | Iya |
| 6 | 3842 | 0.001 | Iya |
| 7 | 3070 | 0.001 | Iya |
| 8 | 3371 | 0.001 | Iya |
| 9 | 3340 | 0.001 | Iya |
| 10 | 3023 | 0.001 | Iya |
| 11 | 2910 | 0.001 | Iya |
| 12 | 3151 | 0.001 | Iya |
| 13 | 3044 | 0.001 | Iya |
| 14 | 3494 | 0.001 | Iya |
| 15 | 3265 | 0.001 | Iya |
| 16 | 3043 | 0.001 | Iya |

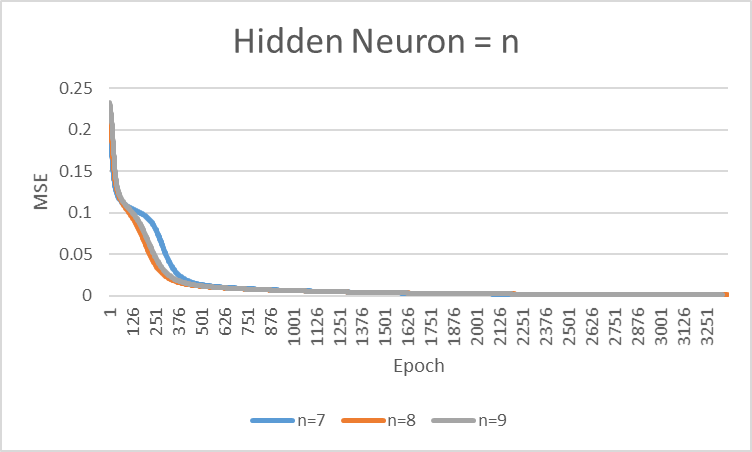
Dari ke-16 percobaan yang dilakukan, terlihat bahwa jumlah *hidden neuron* dapat mempengaruhi hasil pelatihan jaringan saraf tiruan. Jumlah *hidden neuron* berpengaruh terhadap lamanya proses latihan serta konvergensi jaringan terhadap data yang dilatihkan. Proses pelatihan dengan menggunakan *jumlah hidden neuron* yang berbeda menyebabkan terjadinya fluktuasi dalam jumlah iterasi yang harus ditempuh. Pada grafik proses pelatihan dapat dilihat perbedaan proses konvergensi jaringan terhadap data latih, yakni pada Gambar 3, Gambar 4 , Gambar 5, Gambar 6, Gambar 7 Dan Gambar 8 .



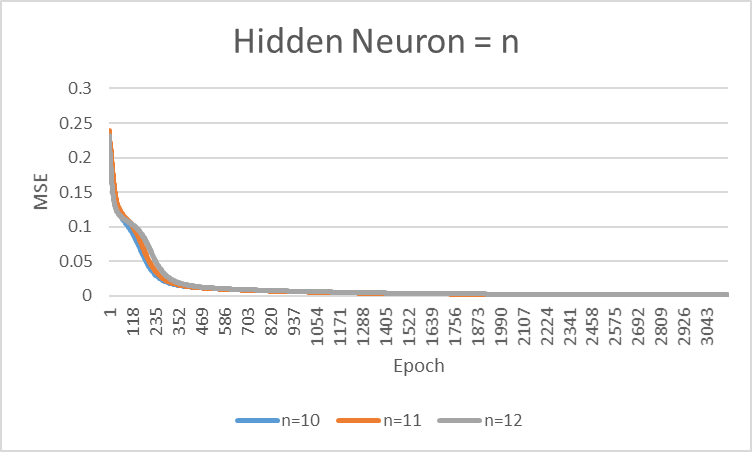
Gambar 5: Grafik Pelatihan Dengan n = 1, 2 dan 3



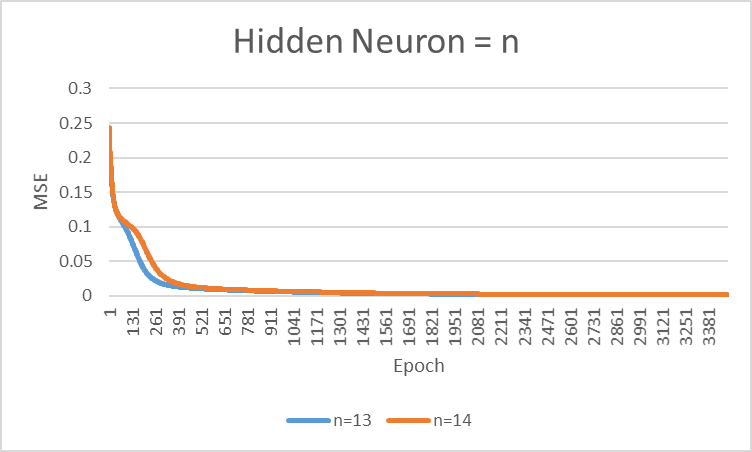
Gambar 6: Grafik Pelatihan Dengan n = 4, 5 dan 6



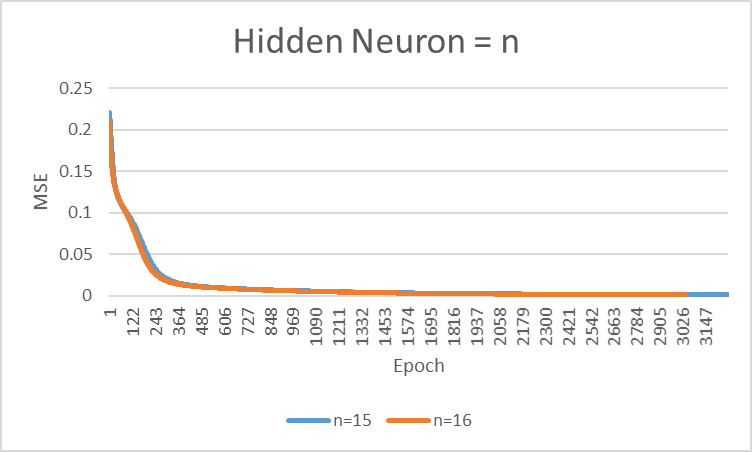
Gambar 7: Grafik Pelatihan Dengan n = 7, 8 dan 9



Gambar 8: Grafik Pelatihan Dengan n = 10, 11 dan 12



Gambar 9: Grafik Pelatihan Dengan n = 13 dan 14



Gambar 10: Grafik Pelatihan Dengan n = 15 dan 16

Dengan menggunakan parameter jaringan saraf seperti diperlihatkan pada Tabel 2 serta jumlah *hidden neuron* yang berbeda-beda dalam setiap percobaan pelatihan yang dilakukan. Penulis melakukan pengujian terhadap sistem yang telah dilatih dengan menggunakan data latih yang terdapat pada Tabel 1. Hasil dari pengujian dari data terlatih ditunjukkan pada Tabel 4 .

Tabel 4: Hasil Pengujian Data Terlatih

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Jumlah Hidden Neuron** | **Jumlah Data** | **Jumlah Dikenali** | **Jumlah Tidak Dikenali** | **Akurasi %** |
| 1 | 30 | 20 | 10 | 66.67 |
| 2 | 30 | 30 | 0 | 100 |
| 3 | 30 | 30 | 0 | 100 |
| 4 | 30 | 30 | 0 | 100 |
| 5 | 30 | 30 | 0 | 100 |
| 6 | 30 | 30 | 0 | 100 |
| 7 | 30 | 30 | 0 | 100 |
| 8 | 30 | 30 | 0 | 100 |
| 9 | 30 | 30 | 0 | 100 |
| 10 | 30 | 30 | 0 | 100 |
| 11 | 30 | 30 | 0 | 100 |
| 12 | 30 | 30 | 0 | 100 |
| 13 | 30 | 30 | 0 | 100 |
| 14 | 30 | 30 | 0 | 100 |
| 15 | 30 | 30 | 0 | 100 |
| 16 | 30 | 30 | 0 | 100 |

Perhitungan akurasi dari hasil pengujian dilakukan dengan menggunakan rumus yang terdapat pada persamaan

|  |  |
| --- | --- |
|  | (6) |

Dari data yang diperlihatkan pada Tabel 4 dapat disimpulkan bahwa seluruh data yang telah dilatih kan dapat dikenali dengan baik oleh sistem. Dari 16 konfigurasi jumlah *hidden neuron* yang dilakukan, hanya konfigurasi menggunakan 1 *hidden neuron* saja yang memberikan hasil yang tidak cukup akurat, yaitu hanya memberikan hasil dengan akurasi sebesar 66.67 %.

Setelah proses pelatihan jaringan saraf dan pengujian data terlatih dilakukan, maka sistem harus diuji dengan menggunakan data yang sebelumnya belum pernah dilatih kan atau dengan data tes. Pada penelitian ini penulis menggunakan 5 data citra berkualitas baik dan 3 data citra berkualitas buruk untuk setiap jenis buah anggur. Setelah proses ekstraksi fitur tekstur dengan metode GLCM dilakukan terhadap data citra tes, maka parameter yang didapatkan dari proses ekstraksi akan digunakan sebagai nilai masukan bagi jaringan saraf *backpropagation* untuk selanjutnya diprediksi atau diklasifikasikan oleh sistem. Data citra tes dari hasil ekstraksi fitur tekstur GLCM yang sudah dinormalisasi ditunjukkan pada Tabel 5.

Tabel 5: Data Citra Tes

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Nama Jenis** | **Entropi** | **Energi** | **Homogenitas** | **Kontras** | **Korelasi** |
| 1 | *Red Globe* | 6.5795 | 0.0041 | 0.3948 | 9.8804 | 0.999 |
| 2 | *Red Globe* | 6.202 | 0.0178 | 0.4229 | 11.2104 | 0.999 |
| 3 | *Red Globe* | 6.6874 | 0.0046 | 0.3849 | 11.2782 | 0.9989 |
| 4 | *Red Globe* | 6.7345 | 0.0035 | 0.3941 | 9.3192 | 0.999 |
| 5 | *Red Globe* | 6.6912 | 0.0049 | 0.3661 | 11.1694 | 0.9987 |
| 6 | *Crimson* | 4.8322 | 0.0966 | 0.5211 | 8.419 | 0.9994 |
| 7 | *Crimson* | 5.4514 | 0.0206 | 0.4755 | 7.6744 | 0.9994 |
| 8 | *Crimson* | 6.0152 | 0.0306 | 0.4543 | 7.972 | 0.9991 |
| 9 | *Crimson* | 6.0367 | 0.0117 | 0.4378 | 9.1942 | 0.9992 |
| 10 | *Crimson* | 5.9226 | 0.009 | 0.4739 | 5.4476 | 0.9995 |
| 11 | *Calmeria* | 3.914 | 0.1619 | 0.7062 | 2.1702 | 0.9992 |
| 12 | *Calmeria* | 4.1457 | 0.1362 | 0.7042 | 2.0694 | 0.9993 |
| 13 | *Calmeria* | 4.3076 | 0.1259 | 0.6903 | 2.3598 | 0.9989 |
| 14 | *Calmeria* | 4.24 | 0.1526 | 0.6959 | 2.5118 | 0.9989 |
| 15 | *Calmeria* | 4.7012 | 0.0646 | 0.6559 | 3.1402 | 0.999 |
| 16 | *Red Globe(Dark)* | 6.4385 | 0.0026 | 0.4511 | 4.8252 | 0.9993 |
| 17 | *Red Globe(Blurr)* | 6.2854 | 0.0065 | 0.4597 | 7.6178 | 0.9994 |
| 18 | *Red Globe(Blurr)* | 6.9979 | 0.0039 | 0.3777 | 11.8944 | 0.9987 |
| 19 | *Crimson(Dark)* | 5.9133 | 0.0039 | 0.4812 | 3.6682 | 0.9995 |
| 20 | *Crimson(Blurr)* | 5.7171 | 0.053 | 0.4569 | 10.5602 | 0.9991 |
| 21 | *Crimson(Blurr)* | 5.4887 | 0.0637 | 0.5416 | 6.928 | 0.9995 |
| 22 | *Calmeria(Dark)* | 6.4024 | 0.0026 | 0.4425 | 4.502 | 0.9986 |
| 23 | *Calmeria(Blurr)* | 4.5337 | 0.0889 | 0.6935 | 2.5438 | 0.999 |
| 24 | *Calmeria(Blurr)* | 4.9876 | 0.0377 | 0.675 | 2.838 | 0.9992 |

Dengan menggunakan data citra tes yang terdapat pada Tabel 5, dilakukan proses prediksi dengan menggunakan konfigurasi jumlah *hidden neuron* yang berbeda seperti halnya yang dilakukan pada data terlatih. Dari beberapa konfigurasi jumlah *hidden neuron* didapatkan hasil terbaik dengan tingkat akurasi mencapai 91.67% menggunakan *hidden neuron* yang berjumlah 16 dan hasil terburuk dengan tingkat akurasi sebesar 62.5% didapatkan pada *hidden neuron* dengan jumlah 1.

Tabel 6: Hasil Pengujian Data Tes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Jumlah Hidden Neuron** | **Jumlah Data** | **Jumlah Dikenali** | **Jumlah Tidak Dikenali** | **Akurasi %** |
| 1 | 24 | 15 | 9 | 62.5 |
| 2 | 24 | 21 | 3 | 87.5 |
| 3 | 24 | 21 | 3 | 87.5 |
| 4 | 24 | 21 | 3 | 87.5 |
| 5 | 24 | 21 | 3 | 87.5 |
| 6 | 24 | 21 | 3 | 87.5 |
| 7 | 24 | 21 | 3 | 87.5 |
| 8 | 24 | 21 | 3 | 87.5 |
| 9 | 24 | 21 | 3 | 87.5 |
| 10 | 24 | 21 | 3 | 87.5 |
| 11 | 24 | 21 | 3 | 87.5 |
| 12 | 24 | 21 | 3 | 87.5 |
| 13 | 24 | 21 | 3 | 87.5 |
| 14 | 24 | 21 | 3 | 87.5 |
| 15 | 24 | 21 | 3 | 87.5 |
| 16 | 24 | 22 | 2 | 91.67 |

Dari data yang diperlihatkan pada Tabel 6 dapat disimpulkan bahwa seluruh data yang diprediksi dapat dikenali dengan baik oleh sistem meskipun data citra tersebut belum pernah dilatihkan sebelumnya. Melakukan proses pengujian terhadap data tes dengan menggunakan konfigurasi jumlah *hidden neuron* yang berbeda, memberikan hasil yang mengindikasikan bahwa jumlah *hidden neuron* dapat mempengaruhi kualitas pelatihan dan juga prediksi dari suatu jaringan saraf tiruan. Dari hasil yang didapat seperti terlihat pada Tabel 6 dapat disimpulkan bahwa konfigurasi jaringan saraf dengan menggunakan jumlah *hidden neuron* 16 dapat memberikan hasil terbaik dibandingkan ke-15 konfigurasi jumlah *hidden neuron* yang lain, yaitu dengan tingkat akurasi mencapai 91.67 %.

**5. PENUTUP**

**5.1. Kesimpulan**

Proses klasifikasi jenis buah anggur dengan menggunakan metode statistik orde kedua bekerja dengan membentuk sebuah matriks *co-occurrecy* dari data citra, dilanjutkan dengan menentukan ciri atau fitur sebagai fungsi dari matriks antara tersebut. Dan fitur yang telah didapat digunakan sebagai masukan bagi jaringan saraf tiruan *backpropagation*. Hasil klasifikasi yang dilakukan oleh sistem mampu memberikan tingkat akurasi yang baik Pada data citra latih sistem mampu mengenali jenis buah dengan tingkat akurasi mencapai 100%, dan pada citra data tes atau data tidak terlatih sistem mampu memberikan kualitas klasifikasi dengan akurasi mencapai 91.67%. Hasil tersebut didapatkan dengan menggunakan konfigurasi jumlah *hidden neuron* sebanyak 16 *neuron*.

**5.2. Saran**

Dari hasil penelitian yang dicapai dirasa masih terdapat kekurangan sehingga memberikan peluang bagi peneliti lain untuk melakukan penelitian lanjutan mengenai topik ini. Peneliti menyarankan untuk melakukan penelitian untuk proses klasifikasi dengan metode ekstraksi fitur yang lebih baik lagi, karena dalam beberapa kasus terdapat kondisi di mana fitur tidak dapat dibedakan meskipun berasal dari objek yang berbeda. Hal ini dapat dilakukan dengan menggabungkan metode statistik orde pertama dan kedua atau menggunakan metode ekstraksi fitur lainnya seperti metode ekstraksi fitur berbasis sudut seperti SIFT, SURF dan ORB.

**Daftar pustaka**

[1] I. A. Halela, “Identifikasi Jenis Buah Apel Menggunakan Algoritma K-Nearest Neighbor ( KNN ) dengan Ekstraksi Fitur Histogram,” UDINUS, 2016.

[2] M. A. Agmalaro, A. Kustiyo, dan A. R. Akbar, “Identifikasi Tanaman Buah Tropika Berdasarkan Tekstur Permukaan Daun Menggunakan Jaringan Syaraf Tiruan,” *J. Ilmu Komput. Dan Agri-Informatika*, vol. 2, 2013.

[3] A. Kadir dan A. Susanto, *Teori Dan Aplikasi Pengolahan Citra*. Yogyakarta: Penerbit ANDI, 2013.

[4] S. R. Arief, “Analisis Tekstur dan Ekstraksi Ciri,” Bandung, 2011.

[5] E. Prasetyo, *Pengolahan Citra Digital dan Aplikasinya Menggunakan Matlab*. Yogyakarta: Penerbit ANDI, 2011.

[6] A. Hermawan, *Jaringan Saraf Tiruan*. Yogyakarta: Penerbit ANDI, 2006.