KLASIFIKASI JENIS GAHARU BERDASARKAN BENTUK DAN WARNA BUAH MENGGUNAKAN METODE *ADAPTIVE*

*NEURO FUZZY INFERENCE SYSTEM* (ANFIS)

**TUGAS AKHIR**

Diajukan Sebagai Salah

Satu Syarat

untuk Memperoleh Gelar Sarjana

Teknik pada Jurusan Teknik Informatika

Oleh

**MUHAMMAD ROFI SAPUTRA**  
**11451105691**



**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU PEKANBARU**

**2019**

**ABSTRAK**

Gaharu merupakan hasil hutan yang dimana memiliki hasil yang serba guna dan memiliki nilai dan daya jual yang tinggi baik untuk para kolektor pohon gaharu maupun masyarakat biasa. Dalam Peraturan Mentri Kehutanan Nomor P.35/Menhut-II/2007, gaharu termasuk dalam daftar 490 jenis Hasil Hutan Bukan Kayu (HHBK) nabati yang potensial untuk dikembangkan. Indonesia merupakan salah satu negara penghasil gaharu yang dimana pulau Kalimantan, Papua, Sumatera dan Kepulauan Nusa Tenggara, merupakan wilayah persebaran alami tanaman penghasil gaharu. Gaharu memiliki 7 genus yang terdiri dari 29 jenis. Riau memiliki 6 jenis kayu gaharu yaitu, gaharu *microcarpa, malacensis, beccariana, crassna, gyrinops* dan *hirta*. Pohon gaharu bisa dikenali jenisnya melalui buah, batang, serta daun. Namun masih banyak orang ataupun petani yang tidak mengetahui cara membedakan jenis pohon gaharu secara kasat mata. Berdasarkan masalah tersebut, maka dilakukan penelitian dalam mengklasifikasikan jenis pohon gaharu berdasarkan warna dan tekstur buah menggunakan metode *Adaptive Neuro Fuzzy Infernce System* (ANFIS). Penelitian ini menggunakan *Hue Saturation Value* (HSV) untuk pengelanan warna dan *Gray Level Co-occurrence Matrix* (GLCM) dalam pengenalan tekstur pada buah.

**Kata Kunci :**  Gaharu, *Adaptive Neuro Fuzzy Infernce System* (ANFIS), *Hue Saturation Value* (HSV), *Gray Level Co-occurrence Matrix* (GLCM).

**BAB** I

**PENDAHULUAN**

**1.1 Latar Belakang**

Pohon Gaharu merupakan hasil hutan yang dimana memiliki hasil yang serba guna dan memiliki nilai dan daya jual yang tinggi baik untuk para kolektor pohon gaharu maupun masyarakat biasa, hal ini dikarenakan hasil gaharu bisa diolah menjadi menjadi bahan dasar wewangian (parfum), kosmetika, minuman, serta menjadi obat herbal (Ismail Muhammad, Muhammad Zin Jusoh, 2018). Dalam Peraturan Mentri Kehutanan Nomor P.35/Menhut-II/2007, gaharu termasuk dalam daftar 490 jenis Hasil Hutan Bukan Kayu (HHBK) nabati yang potensial untuk dikembangkan. Selain itu, gaharu termasuk dalam 5 jenis HHBK yang mendapat prioritas pengembangan, selain jenis rotan, bambu, madu lebah dan sutera (Santoso, 2012).

Indonesia merupakan salah satu negara penghasil gaharu yang dimana pulau Kalimantan, Papua, Sumatera dan Kepulauan Nusa Tenggara, merupakan wilayah persebaran alami tanaman penghasil gaharu. Beberapa jenis yang populer sebagai tanaman penghasil gaharu diantaranya adalah *Aquilaria spp*, *Gyrinops spp* dan *Gonystilus spp.* menurut data hasil Inventaris Hutan Secara Nasional, kerapatan populasi pohon penghasil gaharu Berdasarkan hasil *survey* yang dilakukanoleh Erdy Santoso (2014) di 29 Kabupaten di Jawa Timur diperoleh data 67. 221 pohon penghasil gaharu dari jenis *Gyrinops versteegii*. Data populasi total yang berhasil didapatkan datanya oleh Tim Badan Litbang Kehutanan menunjukkan bahwa jumlah pohon penghasil gaharu yang telah ditanam oleh masyarakat maupun instansi pemerintah di seluruh Indonesia berjumlah: 3.249.959. Jumlah ini belum termasuk pohon penghasil gaharu yang ditanam di Kabupaten lain, yang belum dapat tercatat, misalnya beberapa Kabupaten di propinsi Kalimantan Tengah, Sulawesi Tengah, Sulawesi Utara dan Sulawesi Tenggara, demikian pula di Kabupaten-kabupaten di Propinsi Maluku, NTB dan NTT.

Pohon gaharu bisa dikenali jenisnya dari berbagai macam aspek umum. Seperti batang, isi kulit, daun, bunga, buah, ketinggian dan diameter dari batang pohon. Karakteristik dari bentuk dan warna dari buah menjadi salah satu ciri khas yang dapat membedakan jenis gaharu. Klasifikasi jenis tanaman gaharu melalui buah dapat diamati dan diukur melalui tekstur dan isi dalam buah tersebut. Diantara ciri khas yang dimaksud adalah warna buah serta warna biji, kemudian tekstur buah yang dimana jenis gaharu bisa ditentukan melalui tekstur buah, warna buah serta bentuk buah.

Berdasarkan wawancara dengan Bapak Budi Hasibuan sebagai G*eneral Manager* PT. Plaza Gaharu Indonesia, untuk mengenali jenis pohon gaharu bisa melalui buah, daun, serta batangnya. Klasifikasi jenis tanaman gaharu melalui buah dapat diamati dan diukur melalui tekstur, bentuk dan warna pada buah tersebut. Diantara ciri khas yang dimaksud adalah warna buah, kemudian tekstur buah yang dimana jenis gaharu bisa ditentukan melalui tekstur buah, serta serta bentuk. Berdasarkan kesimpulan yang diambil dari penelitian (Permadi & Murinto, 2015), tekstur pada buah yang dimaksud bisa merupakan garis tekstur pada buah. Dalam pengidentifikasikasian melalui warna buah, dalam pemrosesan warna termasuk didalamnya terdapat ekstraksi informasi tentang *spectral properties* dari permukaan objek dan mencari kesamaan terbaik dari sekumpulan deskripsi yang telah diketahui untuk melakukan pengenalan (Rohmi, Zulfikar, & Gerhana, 2018).

Hasil wawancara yang dilakukan dengan *general manager* gaharu plaza indonesia menyimpulkan bahwa terdapat beberapa kendala dalam mengenali jenis gaharu berdasarkan buah. Kendala ini dikarenakan beberapa buah memiliki kemiripan pada tekstur luar buah, bentuk buah, serta warna pada buah sehingga hal tersebut menyulitkan mahasiswa kehutanan/pertanian yang baru belajar mengenai gaharu dan serta petani baru dari Gaharu dalam mengenali jenis gaharu melalui buah. Dari masalah tersebut, maka dibutuhkan sebuah teknik yang dapat mengenali jenis pohon gaharu berdasarkan buah. Menurut (Setiawan, 2018) Pengolahan Citra Digital merupakan dapat digunakan untuk mengidentifikasi struktur buah melalui bentuk, ukuran, tekstur, serta warna pada buah. Selain itu juga diperlukan ilmu jaringan saraf tiruan untuk mempelajari atau melakukan pendekatan pada pola dengan menerapkan metode neural network (Wiharjo & In, 2010).

beberapa penelitian tentang klasifikasi dan identifikasi jenis tanaman berdasarkan buah dengan pengolahan citra digital telah dilakukan sebelumnya dengan menggunakan berbagai macam metode. Penelitian yang dilakukan oleh (Permana, 2017) tentang aplikasi klasifikasi jenis-jenis buah jeruk dengan metode *K-Nearest Neighbor*. Dari 4 data skenario uji coba, didapatlah akurasi sebesar 63%, 55%, 54%, dan 58%. Penelitian berikutnya oleh (Fuzzy Yustika, Kana Saputra, 2017) melakukan klasifikasi belimbing menggunakan metode *Naïve Bayes,* dengan ekstraksi fitur nilai ­*Red,Green, Blue*. Data citra belimbing yang digunakan berjumlah 120 yang dimana 90 diantaranya merupakan data latih dan 30 merupakan data uji. Hasil klasifikasi menunjukkan hasil akurasi sebesar 80%. Penelitian berikutnya dilakukan oleh (Mega, 2018) tentang aplikasi pengolahan citra untuk klasifikasi buah manga, dimana ekstraksi ciri menggunakaj metode *Learining Vector Quantization* (LVQ) untuk mendapatkan ciri pada setiap karakter masukan buah manga dalam perhitungan nilai RGB. kemudian dilakukan penggabungan untuk menghasilkan vector ciri yang spesifik, selanjutnya dilakukan perhitungan klasifikasi dengan metode *Euclidean Distance*. Berdasarkan hasil uji coba pada aplikasi menghasilkan akurasi sebesar 86.6%.

Dalam melakukan klasifikasi pada buah pohon gaharu dapat dilakukan dengan mengidentifikasi tekstur dan warna pada buah yang dapat dilihat, diamati, serta di ukur. Untuk Ekstraksi tekstur pada buah, tekstur buah bisa diamati dari ukuran buah, ukuran biji, tekstur biji yang dimana beberapa tekstur biji dari buah pohon gaharu ada yang halus serta ada yang berbulu. Beberapa penelitian tentang klasifikasi berdasarkan tekstur seperti penelitian Restu widodo dan agus wahyu (2018) yang melakukan penelitian tentang *Grey Level Co-occurrence Matrix* (GLCM) pada citra buah jeruk keprok untuk menentukan klasifikasi mutu. Pada penelitian ini, dilakukan identifikasi terhadap 100 data uji, yang dimana 60 sebagai data latih dan 40 sebagai data uji. Dari data latih, diambil masing-masing 1 citra baik dan citra buruk berukuran 64x64 piksel. Kemudian dilakukan *pre-processing­* pada citra. Lalu dilakukan pembentukan matriks *Grey Level Co-occurrence Matrix* (GLCM) dan ekstraksi ciri *Grey Level Co-occurrence Matrix* (GLCM) yaitum *contrast, homogeneity, energy, entropy.* Kemudian metode *support Vector Machine* (SVM) digunakan untuk identifikasi citra baik dan citra buruk berdasarkan ciri grade. Penelitian ini menghasilkan tingkat akurasi terbaik sebesar 82,5%. Dari penelitian diatas Maka *Grey Level Co-occurrence Matrix* (GLCM) dipilih sebagai metode ekstraksi fitur untuk klasifikasi jenis buah gaharu

Untuk klasifikasi jenis pohon gaharu berdasarkan buah tidak hanya dilakukan ekstraksi fitur, tetapi juga dilakukan ekstraksi warna. Dalam ekstraksi warna, warna dari buah serta warna dari biji juga menentukan jenis dari pohon gaharu. Salah satu penelitian tentang klasifikasi berdasarkan warna yang pernah diteliti oleh dosen teknik informatika dengan judul penelitian Implementasi Citra dan Klasifikasi K*-Nearest Neighbour* untuk membangun Aplikasi Pembeda Daging Sapi dan Daging Babi untuk membedakan warna pada citra daging babi daging sapi Peneliti menggunakan metode HSV dengan tingkat keakurasian 88,75% (Budianita. E., Jasril., & Handayani, 2015). Berdasarkan penelitian diatas, maka dipilihlah metode HSV atau *Hue Saturation Value* sebagai metode ekstraksi warna pada penelitian ini. Metode *Hue Saturation Value* (HSV) merupakan salah satu model matematis yang merepresentasikan warna dalam angka.

Selain kedua ekstraksi ciri diatas, dilakukan klasifikasi dan penghitungan hitungan akurasi. Beberapa Penelitian tentang klasifikasi serta akurasi pada buah telah dilakukan. Salah satunya penelitian yang dilakukan oleh (Retno Nugroho, Nursinta, 2012) tentang identifikasi buah belimbing menggunakan *Adaptive Neuro Fuzzy Inference System (*ANFIS), yang dimana dilakukan ekstraksi data citra menjadi *Red-Green-Blue* (RGB). Percobaan menghasilkan akurasi 100% untuk kelas asam, 100% untuk kelas sedang, dan 67% untuk kelas manis. Penelitian serupa berikutnya dilakukan oleh (nur sakinah, retno nugroho, 2017) yang mengidentifikasi buah manga gedong gincu Cirebon dengan metode *Adaptive Neuro Fuzzy Interference System* (ANFIS), dengan metode ekstraksi *Red-Green-Blue* (RGB). Pada pelatihan ANFIS menggunakan algoritma *hybrid* dengan metode  *Recursive Least Square Estimator*  (RLSE), dan jumlah *Epoch* sebanyak 1000, nilai error 0.5220 dan mampu mengidentifikasi jenis buah Mangga Gedong Gincu Cirebon dengan tingkat akurasi sebesar 66,6%.

Berdasarkan latar belakang tersebut, akan dilakukan tugas penelitian tentang identifikasi jenis gaharu berdasarkan buah dengan metode klasifikasi *Adaptive Neuro Fuzzy Inference System* (ANFIS), penelitian ini diterapkan kedalam pengolahan citra digital dengan menggunakan ekstraksi fitur tekstur GLCM dan ekstraksi warna HSV.

1. **Rumusan Masalah**

Berdasarkan uraian latar belakang diatas, maka didapatkan rumusan masalah yaitu :

1. Bagaimana menerapkan metode ekstraksi fitur tekstur dengan GLCM dan ekstraksi fitur warna dengan HSV pada buah pohon Gaharu.
2. Bagaimana menerapkan metode ANFIS untuk identifikasi jenis pohon gaharu berdasarkan buah
3. **Tujuan Penelitian**

Tujuan yang ingin dicapai pada penelitian ini, yaitu :

1. Menerapkan metode ekstraksi fitur tekstur denga GLCM dan ekstraksi fitur warna dengan HSV pada buah pohon Gaharu.
2. Menerapkan metode ANFIS untuk klasifikasi pohon gaharu berdasarkan buah.
3. Mengukur tingkat akurasi penerapan metode GLCM, HSV, dan ANFIS dalam mengklasifikasi jenis pohon Gaharu berdasarkan buah.
4. **Batasan Masalah**

Batasan yang diperlukan agar pembahasan tidak terlalu luas, maka dari itu penulis membatasi masalah sebagai berikut:

1. Klasifikasi dan identifikasi jenis pohon gaharu hanya dilakukan pada 6 jenis pohon yaitu Gaharu Candan (*Aquilaria Microcarpa*), Gaharu Ahir (*Aquilaria Malaccensis)*, Gaharu Candan Gajah (*Aquilaria* *Beccarian*a), Gaharu *Crassna*, Gaharu *Gyrinops*, dan Gaharu Karas Baldu (*Aquilaria hirta*).
2. Parameter hasil ekstraksi fitur HSV terdiri dari *Hue, Saturation, Value.* Sedangkan pada GLCM terdiri dari *ASM, Contrast, Korelasi, IDM, dan Entrophy*.
3. Buah yang klasifikasikan ialah buah bagian luar.
4. **Sistematika Penulisan**

Sistematika penulisan laporan tugas akhir ini dibagi menjadi beberapa bab yaitu:

**BAB I PENDAHULUAN**Bab ini menjelaskan dasar-dasar dari penulisan laporan tugas akhir yang terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, dan sistematika penulisan laporan tugas akhir.

**BAB II LANDASAN TEORI**

Bab ini menjelaskan tentang teori pendukung baik dari buku, jurnal, atau penelitian terkait yang terdiri dari pengolahan citra digital, ekstraksi gitur GLCM, fitur morfologi digital, ANFIS, dan pohon Gaharu.

**BAB III METODOLOGI PENELITIAN**

Bab ini menguraikan tentang tahapan-tahapan yang dilakukan selama melakukan penelitian ini yaitu studi pustaka, perumusan masalah, pengumpulan data, analisa dan perancangan, implementasi dan pengujian, serta kesimpulan dan saran.

**BAB IV ANALISA DAN PERANCANGAN**

Bab ini berisi tentang uraian analisa metode yang digunakan dan perancangan sistem yang akan dibangun pada penelitian tugas akhir ini.

**BAB V IMPLEMENTASI DAN PENGUJIAN**

Bab ini berisi tentang penjelasan hasil dari analisa dan perancangan yang di implementasikan ke dalam sistem serta melakukan pengujian terhadap metode yang telah digunakan untuk mendapatkan hasil akurasi.

**BAB VI KESIMPULAN DAN SARAN**Bab ini berisi tentang kesimpulan dan saran agar penelitian yang dilakukan dapat dikembangkan menjadi penelitian yang lebih baik.

**BAB II  
LANDASAN TEORI**

1. **Gaharu**

Gaharu merupakan salah satu hasil hutan bukan kayu yang mempunyai nilai ekonomi, budaya dan sosial yang sangat tinggi. Gaharu termasuk dalam 490 jenis Hasil Hutan Bukan Kayu (HHBK) nabati yang potensial untuk dikembangkan (Permenhut, 2007). Gaharu mememiliki manfaat yang dimana pohon gaharu mengandung essens yang disebut sebagai minyak essens (*essential oil*) yang dapat dibuat dengan ekstraksi atau penyulingan dari gubal gaharu.Essens gaharu ini digunakan sebagai bahan pengikat (*fixative*) dari berbagai jenis parfum, kosmetika, dan obat-obatan herbal.Selain itu, serbuk atau abu dari gaharu dapat digunakan sebagai bahan pembuatan dupa/hio dan bubuk aroma therapy, dan daun pohon gaharu bisa dibuat menjadi teh yang dapat membantu kebugaran tubuh. Banyaknya manfaat serta tingginya permintaan dan harga jual gaharu di pasar internasional serta semakin langkanya tanaman penghasil gaharu di hutan alam, pada sisi lain telah mendorong masyarakat di berbagai daerah melakukan budidaya tanaman pengahasil gaharu. saat ini penanaman atau budidaya tanaman penghasil gaharu oleh masyarakat, kelompok tani, swasta serta instansi pemerintah telah banyak dilakukan di berbagai wilayah/kabupaten di seluruh Indonesia (Santoso, 2012).

Saat ini diperkirakan terdapat lebih kurang 27 jenis tumbuhan penghasilkan gaharu yang dikelompokkan ke dalam delapan marga dan tiga suku. Bentuk hidupan tumbuhan penghasil gaharu dapat berupa pohon, semak dan perdu yang merambat. Menurut (Wijoyono, 2002) di Indonesia terdapat 8 jenis pohon penghasil kayu gaharu yaitu *Aetoxylon sympetalum*, *Aquilaria hirta*, *Aquilaria malaccensis, gonysytylus bancamus, gonysytylus macrophyllus, enkleia malaccensis, wiksroemia polyantha,* dan *wikstroemia tenuiramis*. Semua jenis tersebut merupakan anggota suku Thymeleacceae. Selain itu, ada satu marga lan yang juga menghasilkan gaharu yaitu *Gyrinops*. Menurut (whitmore T.C, 1989) *Gyrinops* dapat dijumpai di daerah Kawasan Timur Indonesia (Sulawesi, Nusa tenggara, dan Papua). Tumbuhan penghasil gaharu umumnya tumbuh di beberapa pulau yaitu, Kalimantan (12 jenis), sumatera (10 jenis), nusa tenggara (3 jenis), Papua (2 jenis), jawa (2 jenis), dan maluku (1jenis). Sedangkan di Provinsi Riau sendiri terdapat 6 jenis pohon gaharu yaitu Gaharu jenis *Microcarpa*, *Malacensis*, *Beccariana* (candan gajah), *Crassna*, *Gyrinops*, dan *Hirta*.

1. *Aquilaria Microcarpa*

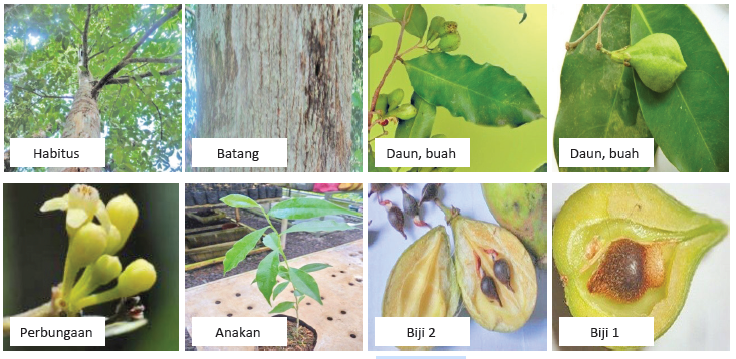
Habitus pohon tinggi hingga 40 m, diameter 80 cm. Batang tidak lurus, kulit batang beralur, warna coklat kelabu. Daun bundar telur lanset-lanset terbalik, tipis, ukuran 4-11 x 1-4 cm, ujung lancip-luncip, pangkal daun bentuk pasak-tirus, tepi rata dan menebal; warna hijau, permukaan atas licin, bawah kadang berbulu halus, panjang tangkai 3-5 mm, berbulu; tulang daun sekunder sejajar, agak rapat, jumlah 10-19 pasang, terlihat jelas menonjol pada permukaan bawah. Perbungaan bentuk payung, muncul diujung ranting, bawah ketiak daun dan di atas ketiak tangkai, bercabang, masing-masing 5-11 bunga, panjang tangkai perbungaan 5-10 mm; bunga bentuk tabung panjang 4-5 mm, warna putih kekuningan, panjang tangkai 5 mm, bagian luar berbulu halus, bagian dalam berbulu jarang. Buah kapsul, licin, berukuran 1-1,5 x 1 cm, ujung buah tumpul, panjang kelopak buah 0,5 cm, daging buah tebal dan berbulu, panjang tangkai buah 0,7-1 cm. Biji bulat telur, hitam, ukuran 6-4 mm, berbulu tebal, warna kecoklatan, jumlah biji 1-2, pangkal biji terdapat sumbat lembaga warna putih. Anakan jenis ini, bentuk daunnya jorong sampai lanset, tersusun berseling (Adi, Titi, & Erdy, 2014).



Gambar 2.1 *Aquilaria Microcarpa*

1. *Aquilaria Malaccensis*

Habitus pohon tinggi 25-50 m, diameter 60 cm. Batang tegak, lurus, kadang berbanir, kulit batang licin, beretak tipis, warna coklat kelabu, kulit dalam putih, kayu gubal putih kekuningan (coklat muda). Kayu mengandung resin (Kosmiatin, Husni dan Mariska, 2005 *dalam* Roro, 2011). Daun bundar telur-lonjong, tipis tidak berbulu, ukuran 5-14 x 2,5-5 cm, ujung luncip, pangkal lancip, tirus, tumpul, tepi bergelombang, warna daun hijau tua, permukaan bawah hijau terang, kadang berbulu, panjang tangkai 4-6 mm dan berbulu, tulang daun sekunder menyirip tidak teratur, jumlah 12-16 pasang, terlihat jelas menonjol di permukaan atas, tulang daun permukaan bawah berbulu halus. Perbungaan bentuk payung, muncul di ujung ranting, bawah ketiak daun dan di atas ketiak tangkai, bercabang 2-3, masing-masing cabang 10 bunga, panjang tangkai perbungaan 5-15 mm. Bunga bentuk tabung, panjang 5-6 mm, warna hijau kekuningan, panjang tangkai bunga 3-6 mm, tabung bunga bagian dalam tidak berbulu dan bagian luar berbulu. Buah kapsul, licin, bulat telur sungsang, ukuran 2,5-3,5 x 2,5 cm, ujung buah tumpul dan pangkal buah menyempit, daging buah tebal tidak berbulu panjang tangkai buah 1 cm. Biji bentuk bulat telur, hitam, berukuran 10 x 6 mm, bagian pangkal biji bengkok seperti ekor berbulu lebat, warna merah, jumlah biji 1-2. Anakan jenis ini, bentuk daun jorong sampai lanset, tersusun berseling (Adi et al., 2014).



Gambar 2.2 *Aquilaria Malaccensis*

1. *Aqularia Beccariana*

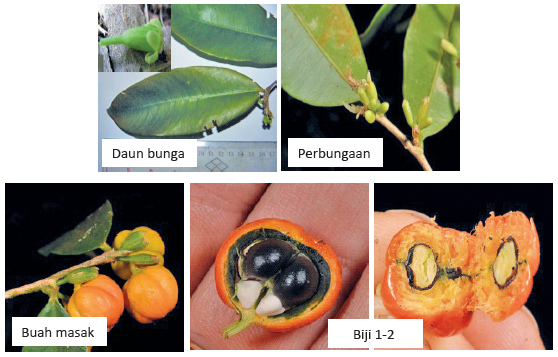
Habitus pohon besar, tinggi 28-40 m, diameter 36-60 cm. Batang berkulit tipis, beralur, warna coklat kelabu, berserat panjang yang sangat kuat sehingga sering dimanfaatkan untuk tali. Daun bentuk jorong- lonjong, tipis, ukuran (7-) 11-27 x (3-) 6-8,5 cm, ujung luncip, pangkal runcing - tumpul, tepi menebal; warna hijau, kedua permukaan daun licin, kadang bulu tersebar pada bagian bawah, panjang tangkai 5-7 mm; tulang daun sekunder berjumlah (10-) 15-25 pasang, menonjol jelas pada permukaan bawah. Perbungaan bentuk payung, muncul diujung ranting, ketiak daun, panjang tangkai perbungaan 5-15 mm. Bunga bentuk tabung memanjang sekitar 1 cm, warna hijau kekuningan/putih kekuningan, tangkai bunga 3-7 mm, bagian luar berbulu jarang dan bagian dalam berbulu. Buah bentuk gelendong dan gepeng, ukuran 2-3,5 x 1 cm, berkulit tipis, menyempit pada kedua ujungnya, di bagian tengah sedikit berlekuk, panjang tangkai buah 1 cm. Biji bulat telur, warna hitam, ukuran 10-5 mm, berbulu lebat warna coklat kemerahan, jumlah biji 1-2 dalam satu buah (Adi et al., 2014).



Gambar 2.3 *Aquilaria Bacceriana*

1. *Gyrinopsis cumingiana*

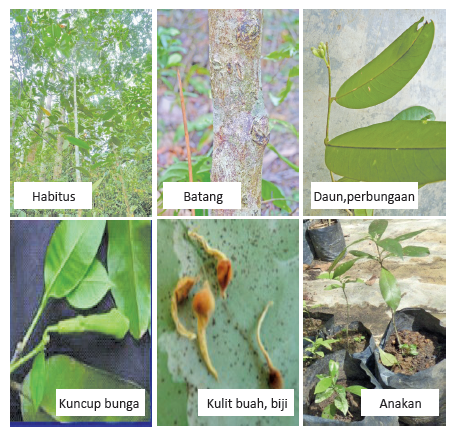
Habitus pohon kecil-sedang, tinggi 5-20 m, diameter hingga 40 cm. Batang luar coklat kelabu, dengan bintik-bintik halus, berserat panjang yang sangat kuat sehingga dimanfaatkan untuk tali. Daun lonjong-lanset, jorong-lonjong atau bundar telur-lonjong, ukuran 14-18 x 2-8,5 cm, tersusun berseling, pangkal menyempit, ujung luncip, warna hijau, kedua permukaan daun licin; tulang daun sekunder berjumlah 12-18 pasang, terlihat jelas pada permukaan bawah, panjang tangkai daun 4-6 mm. Perbungaan bentuk payung, muncul pada ketiak daun dan menempel pada batang dengan jumlah sangat banyak, tangkai perbungaan 3 mm. Bunga bentuk tabung, warna hijau, panjang bunga 13-16 mm, tabung bunga bagian luar berbulu lebat dan di dalam tidak berbulu, biasanya terdapat bintik-bintik kelenjar seperti kutil yang tersebar. Buah bulat sedikit bulat telur sungsang atau jorong, berlekuk 4, keriput, buah muncul dari celah lateral tabung bunga, warna hijau berubah kuning pada waktu matang, berukuran sekitar 1,5-2 cm. Biji bulat telur atau bundar pipih, jumlah 1-2 biji, pada bagian pangkal terdapat sumbat lembaga warna putih. Tidak ditemukan anakan (Adi et al., 2014).



Gambar 2.4 *Gyrinopsis Cumingiana*

1. *Aquilaria Hirta*

Habitus pohon kecil-sedang, tinggi hingga 15 m, diameter 17 cm. Batang tegak, lurus, warna kelabu, berkulit tipis dengan serat panjang dan kuat, ranting berbulu halus lebat. Daun bentuk bundar telur melebar-lonjong, jorong-lonjong, mirip daun *A. beccariana,* berseling, hijau kusam, berukuran 15-16 cm x 8-10 cm, pangkal daun membundar, ujung daun luncip; tulang daun sekunder 16-30 pasang dan terlihat jelas pada permukaan bawah; panjang tangkai 5-7 mm. Perbungaan bentuk payung, muncul di ketiak daun dekat ujung ranting, jumlah 5-14 bunga, panjang perbungaan 10 mm. Bunga bentuk tabung, panjang 1 cm, mekar tidak beraturan, berbulu rapat, putih/kuning gading, tangkai bunga 2 cm. Bakal buah berbulu lebat. Buah bentuk lanset sungsang atau gepeng dan gelendong, membesar keujung dan menyempit kepangkal, ukuran 3,5-5 x 1 cm, berbulu halus dan rapat, warna keemasan, kulit buah tipis; buah muncul dari celah lateral tabung bunga. Biji bentuk bulat telur, berukuran 10 x 6 mm, berbulu, ujung biji berparuh pada pangkal biji bentuk pasak, panjang 10 mm, warna hitam mengkilap berukuran panjang 2 cm, jumlah biji 1. Anakan jenis ini, bentuk daunnya jorong dan tersusun berhadapan (Adi et al., 2014).



Gambar 2.5 *Aquilaria hirta*

1. **Pengolahan Citra Digital**

Pengolahan citra digital adalah sebuah disiplin ilmu yang mempelajari hal-hal yang berkaitan dengan perbaikan kualitas gambar (peningkatan kontras, transformasi warna, restorasi citra), transformasi gambar (rotasi, translasi, skala, transformasi geometrik), melakukan pemilihan citra ciri (*feature images*) yang optimal untuk tujuan analisis, melakukan proses penarikan informasi atau deskripsi objek atau pengenalan objek yang terkandung pada citra, melakukan kompresi atau reduksi data untuk tujuan penyimpanan data, transmisi data, dan waktu proses data. Input dari pengolahan citra adalah citra, sedangkan output-nya adalah citra hasil pengolahan (Sutoyo, 2009).

1. Konversi Jenis Citra

Jenis Konversi citra yang digunakan dalam pemrosesan citra adalah citra berwarna, citra berskala keabuan (*grayscale*), dan citra biner. Citra berwarna adalah jenis citra yang menyajikan warna dalam bentuk komponen *Red, Green,* dan *Blue* atau disingkat (RGB) yang dimana masing-masing komponen warna menggunakan 8 bit (nilainya berkisar 0 sampai dengan 255). Dengan demikian, warna yang bisa disajikan mencapai 255 x 255 x 255 atau 16.581.375 warna. Citra berskala keabuan (*grayscale*) menangani gradasi warna hitam dan putih yang menghasilkan efek warna abu-abu, yang dinyatakan dengan intensitas berkisar antara 0 dan 255, di mana 0 menyatakan hitam dan 255 menyatakan putih. Citra biner adalah citra dengan setiap piksel yang hanya dinyatakan dengan dua kemungkinan nilai yaitu 0 menyatakan warna hitam dan 1 menyatakan warna putih (Kadir, A. Susanto, 2013).

1. Ekstraksi Fitur

Ekstraksi Fitur merupakan tahapan mengekstrak ciri atau informasi dari objek di dalam citra yang ingin dikenali atau dibedakan dengan objek lain. Salah satu ekstraksi fitur yang umum digunakan untuk mengekstraksi fitur adalah ekstraksi fitur warna dan ekstraksi fitur tekstur.

1. Ekstraksi Fitur Warna

Untuk membedakan suatu objek dengan warna tertentu dapat menggunakan nilai *hue* yang merupakan representasi dari cahaya tampak (merah, jingga, kuning, hijau, biru, ungu). Nilai hue dapat dikombinasikan dengan nilai *saturation* dan *value* yang merupakan tingkat kecerahan suatu warna. Untuk mendapatkan ketiga nilai tersebut, perlu dilakukan konversi ruang warna citra yang semula RGB (Red, Green, Blue) menjadi HSV (Hue, Saturation, Value).

1. Ekstraksi Fitur Tekstur

Fitur tekstur banyak digunakan sebagai fitur untuk temu kembali citra. Hal ini karena beberapa objek mempunyai pola-pola tertentu yang bagi manusia mudah untuk dibedakan. Aplikasi tekstur dibagi menjadi dua kategori yaitu untuk kepentingan segmentasi dan klasifikasi tekstur (Kadir, A. Susanto, 2013).

1. ***Hue Saturtion Value* (HSV)**

HSV merupakan singkatan dari *Hue, Saturation*, dan *Value* yang menggambarkan warna 3 dimensi, dimana HSV ini berusaha untuk memperbaiki model warna RGB dan menghubungkan antar warna. Jika HSV di gambarkan sebagai roda, maka yang akan terjadi sumbu pusat bergerak dari warna putih yang berada di atas ke warna hitam yang berada di bawah dengan warna-warna netral lain berada diantaranya (Purnamasari, 2013).

Untuk mendapatkan nilai H, S, V berdasarkan R, G, dan B, terdapat beberapa cara. Cara yang tersederhana (Acharya & Ajoy K, 2005) adalah seperti berikut.

(2.1)

(2.2)

(2.3)

Namun, cara diatas membuat *hue* tidak terdefinisikan jika S bernilai nol. Cara kedua terdapat pada (Acharya & Ajoy K, 2005). Rumus-rumus yang digunakan sebagai berikut:

, (2.4)

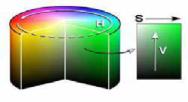
(2.5)

(2.6)

(2.8)

(2.9)

Berikut ini merupakan model warna *hue saturation value* (HSV), dapat dilihat pada gambar 2.1:



**Gambar 2.1 Model warna HSV**

Nilai HSV yang di normalisasi ke dalam angka yang lebih sederhana adalah nilai yang menjadi tolak ukur pada HSV agar mempersingkat waktu komputasi tanpa mengurangi ke akurasiannya (Syahid, 2016). Untuk mengkonversi nilai RGB ke dalam bentuk HSV dapat dilakukan dengan persamaan sebagai berikut:

Model warna HSV terbagi menjadi 3 bagian yaitu *hue, saturation,*dan *value*

1. *Hue*

*HSV* mempresentasikan *hue* sebagai warna yang memiliki nilai warna antara0-360 derajat.

1. *Saturation*

*Saturation* yang semakin rendah maka akan mengakibatkan warna tersebutpudar, jika nilai saturation bernilai 0 maka wana nya akan pudar dan jika 1 maka nilainya akan menunjukkan warna utama pada warna tersebut karena *color space* pada saturation mengindikasi warna abu abu dari 0-100%.

1. *Value*

*Value* yang bernilai 0 maka dia akan menunjukkan warna hitam karenasemakin rendah *value* maka kecerahan value akan semakin gelap dan jika warna *value* menunjukkan warna 100 maka dia akan mendekati warna putih karena *value* menunjukkan kecerahan pada warna.

Setelah mendapatkan nilai HSV dilakukan perhitungan nilai *mean* dengan rumus statis rerata untuk proses klasifikasi yang akan dilakukan selanjutnya (Kadir, A. Susanto, 2013).

1. ***Gray Level Co-Occurrence Matrix* (GLCM)**

Gray Level Co-occurrence Matrix (GLCM) adalah suatu metode yang digunakan untuk analisis tekstur/ekstraksi ciri. GLCM merupakan suatu matriks yang menggambarkan frekuensi munculnya pasangan dua piksel dengan intensitas tertentu dalam jarak dan arah tertentu dalam citra (Eko, 2011). Untuk mendapatkan fitur pada GLCM, perlu beberapa besaran yang digunakan. Lima besaran yang digunakan untuk GLCM diantaranya *angular second moment* (ASM), Kontras, *inverse different moment* (IDM), *entropy,* dan *correlation* (Ratnasari, 2017).

1. ASM (*angular second moment*)

Parameter ASM mengatur keseragaman. ASM merupakan ukuran homogenitas citra yang dihitung dengan rumus sebagai berikut.

(2.10)

1. Kontras (*Contrast*)

Parameter dari kontras bertugas untuk mengukur perbedaan moment. Kontras merupakan ukuran keberadaan variasi aras keabuan piksel dengan rumus yang dihitung dengan cara sebagai berikut.

(2.11)

1. IDM (*inverse different moment*)

IDM digunakan untuk mengukur homogenitas. IDM bersifat sensitive dengan nilai seitaran diagonal utama yang dihitung dengan rumus sebagai berikut.

(2.12)

1. Entropi (*entropy*)

Entropi merupakan ukuran keacakan aras keabuan didalam citra yang dimana parameter ini akan bernilai tinggi jika citra yang dimiliki tidak seragam 1 sama lain. Entropi dihitung dengan rumus sebagai berikut.

(

1. Korelasi (*correlation*)

Korelasi merupakan parameter GLCM untuk mengukur liniearitas dari sejumlah pasangan piksel yang dihitung dengan rumus sebagai berikut.

(2.14)

Dengan

(2.15)

(2.16)

(2.17)

(2.18)

Tahapan yang dilakukan pada perhitungan GLCM adalah sebagai berikut:

1. Pembentukan matriks awal GLCM dari pasangan dua piksel yang berjajar sesuai dengan arah 0°, 45°, 90° atau 135°.
2. Membentuk matriks yang simetris dengan menjumlahkan matriks awal GLCM dengan nilai transposnya.
3. Menormalisasi matriks GLCM dengan membagi setiap elemen matriks dengan jumlah pasangan piksel.
4. **Normalisasi Data**

Normalisasi data adalah proses mengubah atau mentransformasikan data menjadi linear dengan rentang nilai dari 0 sampai 1 tanpa kehilangan karakteristik dari data tersebut. Normalisasi data adalah proses transformasi data menjadi linier dengan rentang nilai dari 0 sampai 1 tanpa kehilangan karakteristik dari data tersebut. Proses normalisasi data menggunakan rumus sebagai berikut.

(2.19)

Keterangan:  
 = Nilai setelah normalisasi

= Nilai sebelum normalisasi

min(*X*) = Nilai minimum

max(*X*) = Nilai maksimum

1. **Jaringan Syaraf Tiruan**

Jaringan Syaraf Tiruan (JST) adalah sistem komputerisasi sebagai pemroses informasi yang memiliki karater mirip dengan jaringan syaraf biologi pada saat menangkap informasi dari ‘dunia luar’. Maksud sebenarnya dari JST adalah berusaha membuat sebuah model sistem komputasi informasi yang dapat menirukan rangkaian cara kerja jaringan syaraf biologis. Secara sederhana, JST adalah sebuah alat pemodelan [data](https://id.wikipedia.org/wiki/Data) [statistik](https://id.wikipedia.org/wiki/Statistik) non-linier. JST dapat digunakan untuk memodelkan hubungan yang kompleks antara input dan output untuk menemukan pola-pola pada data. Menurut suatu teorema yang disebut "teorema penaksiran universal", JST dengan minimal sebuah lapis tersembunyi dengan fungsi aktivasi non-linear dapat memodelkan seluruh fungsi terukur Boreal apapun dari suatu dimensi ke dimensi lainnya (Kurt, 1989).

1. **Fuzzy *Logic***

Logika *fuzzy* adalah cabang dari sistem kecerdasan buatan (*Artificial Inteligent*) yang mengemulasi kemampuan manusia dalam berfikir ke dalam bentuk algoritma yang kemudian dijalankan oleh mesin. Algoritma ini digunakan dalam berbagai aplikasi pemrosesan data yang tidak dapat direpresentasikan dalam bentuk biner. Logika *fuzzy* menginterpretasikan statemen yang samar menjadi sebuah pengertian yang logis. logika *fuzzy* tidak mutlak 1 dan 0, namun hasilnya berkisar antara 0 sampai dengan 1. Jadi logika *fuzzy* tidak hanya menghasilkan nilai benar dan salah saja, tapi dapat berisi setengah benar, semperempat salah dan seterusnya (Kusumaewi, 2000).

(2.20)

1. ***Fuzzy Inference System* (FIS)**

*Fuzzy Inferense System* (FIS) atau *Fuzzy Inferense Engine* adalah sistem yang dapatmelakukan penalaran dengan prinsip serupa seperti manusia melakukan penalaran dengan nalurinya (Alavi, et al., 2010). Langkah pertama dari FIS adalah untuk menetapkan nilai keanggotaan untuk data input dan output (Alidoosti, et al., 2012).

1. **Fungsi Keanggotaan Logika Fuzzy**

Fungsi Keanggotaan (*Membership Function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik input data (sumbu x) kepada nilai keanggotaannya (sering juga disebut derajat keanggotaannya) yang mempunyai interval mulai 0 sampai 1.Terdapat bermacam jenis kurva yang dapat digunakan untuk mempresentasikan himpunan *fuzzy.* Semua kurva yang memiliki tinggi maksimum 1 (satu) dapat digunakan untuk mempresentasikan suatu himpunan *fuzzy*. Suatu fungsi keanggotaan dicirikan oleh parameter fungsi keanggotaan . Berikut 7 fungsi keanggotaan pada logika *fuzzy*.

1. Representasi Linier

Penggambaran *input* ke derajat keanggotaannya digambarkan sebagai suatu garis lurus pada representasi linier.

1. Representasi Kurva Segitiga

Gabungan antara dua garis lurus (linier) yang memiliki titik temu disebut sebagai kurva segitiga.

1. Representasi Kurva Trapesium

Pada dasarnya kurva trapesium ini memiliki bentuk seperti segitiga namun, pada kurva trapesium ini ada beberapa titik yang memiliki nilai keanggotaan satu [1].

1. Representasi Kurva Bentuk Bahu

Bentuk segitiga yang berada pada bagian tengah kurva bentuk bahu memiliki fungsi untuk mempresentasikan suatu variabel.

1. Representasi Kurva Gauss

Kurva gauss ini memiliki bentuk yang mirip dengan kurva lonceng. Variabel pada fungsi keanggotaan ini dicirikan oleh parameter *c* dan .

1. Represntasi kurva *sigmoid*

Pertumbuhan pada kurva *sigmoid* ini bergerak dari sisi paling kiri dengan nilai keanggotaan nol [0] ke sisi paling kanan dengan nilai keanggotaan satu [1].

1. Representasi Kurva Lonceng (*Generalized Bell*)

Kurva lonceng memiliki ciri pada parameter *a, b,* dan *c*. Parameter *b* bernilai positif dan parameter *c* menyatakan nilai tengah (mean).

1. **Adaptive Neuro Fuzzy (ANFIS)**

ANFIS adalah penggabungan mekanisme *fuzzy inference system* yang digambarkan dalam arsitektur jaringan syaraf. Sistem inferensi *fuzzy* yang digunakan adalah sistem inferensi fuzzy model Tagaki-Sugeno-Kang (TSK) orde satu dengan pertimbangan kesederhanaan dan kemudahan komputasi. Salah satu contoh ilustrasi mekanisme inferensi fuzzy TSK orde satu dengan dua masukan x1 dan x2 (Syamsuar, 2014).

Misalkan ada 2 input x dan y, serta 1 output z maka ada 2 aturan berikut :

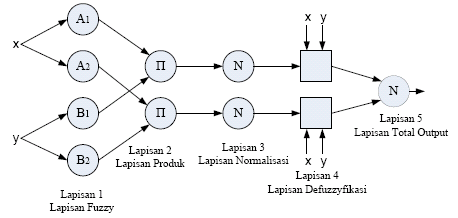
Rule 1 : (2.21)

Rule 2 : (2.22)

Mekanisme *fuzzy inference* *system* pada metode ini adalah

(2.23)

Pada Jaringan ANFIS terdapat 2 masukan yaitu x, ydan 1 keluaran z. Jaringan ANFIS terdiri dari beberapa lapisan yang ditunjukan pada Gambar 2.2.



**Gambar 2.2 Struktur Anfis**

Dari gambar diatas dapat diketahui bahwa arsitektur *Adaptive Neuro* *Fuzzy Inference System (ANFIS)* terdiri atas lima lapisan, dan setiap lapisterdapat node/simpul. Terdapat dua macam node yaitu node adaptif (bersimbol kotak) dan node tetap (bersimbol lingkaran). Fungsi dari setiap lapis adalah sebagai berikut :

1. Pada lapisan 1 berfungsi sebagai proses fuzzyfikasi. Output dari node I pada lapisan 1 dinotasikan sebagai O1,i. Setiap node 1 pada lapis 1 ini adalah node adaptif (parameter dapat berubah) dengan fungsi node sebagai berikut

*O1,i = μAi (x1) untuk I = 1, 2* (2.24)

*O1.i = μBi-2 (x2) untuk i = 3,4 (2.25)*

tiap-tiap input tersebut dibagi menjadi 2 fungsi keanggotaan, x1 dibagi dalam A1 dan A2.

1. Pada Lapisan 2 Semua Node adalah nonadaptive (parameter tetap) fungsi node ini adalah mengalihkan sinyal masukan yang datang.

*O2,i = wi = μAi (x1) μBi (x2), i = 1, 2 (2.26)*

1. Pada lapisan 3, Lapisan Setiap node adalah node nonadaptif yang menampilkan fungsi derajat pengaktifan ternomalisasi (*normalized firing* *strength*) yaitu rasio keluaran node ke-i pada lapisan sebelumnya terhadapseluruh keluaran lapisan sebelumnya, dengan bentuk fungsi node.

= (2.27)

1. Pada Lapisan 4, setiap nodenya bersifat adaptif dengan fungsi node

= (2.28)

adalah keluaran atau *output* dari *layer* 3 dan *f* adalah parameter konsekuen pada *fuzzy* model Sugeno orde nol.

1. Pada lapisan 5ini hanya ada satu node tetap yang fungsinya untuk menjumlahkan semua masukan. Fungsi simpul, *Node* pada *layer* 5 ini dinotasikan dengan Σ.

(2.29)

Jaringan adaptif dengan kelima lapisan tersebut ekivalen dengan sistem inferensi fuzzy model Sugeno nol.

1. **Pengujian Akurasi**

Pengukuran terhadap suatu kinerja sistem klasifikasi adalah satu hal yang penting. Sistem klasifikasi memberikan informasi seberapa baik dalam mengklasifikasikan data.

* 1. ***Confusion Matrix***

*Confusion matrix* merupakan salah satu metode yang digunakan untuk mengukur suatu kinerja dari hasil sistem klasifikasi. *Confusion matrix* bekerja dengan membandingkan hasil klasifikasi suatu kinerja sistem dengan hasil klasifikasi lainnya. Terdapat 4 jenis dalam sistem klasifikasi yaitu, klasifikasi *binary, multi-class, multi-label,* dan *hierarchical* (Prasetyo & Agustin, 2011).

Pada jenis klasifikasi *binary* hanya memiliki 2 keluaran kelas.

**Tabel 2.1 Klasifikasi *Binary***

|  |  |  |
| --- | --- | --- |
| **Kelas** | **Terklasifikasi Positif** | **Terklasifikasi Negatif** |
| **Positif** | TP ( True Positive ) | FN ( False Negative ) |
| **Negatif** | FP ( False Positive ) | TN ( True Negative ) |

Berdasarkan nilai terklasifikasi maka, diperolehlah nilai akurasi, presisi dan *recall*.

(2.30)

(2.31)

(2.32)

1. **Penelitian Terkait**

Berikut Merupakan Tabel 2.1 tentang penelitian yang terkait dengan klasifikasi buah dan metode yang digunakan seperti ekstraksi GLCM dan HSV serta ANFIS.

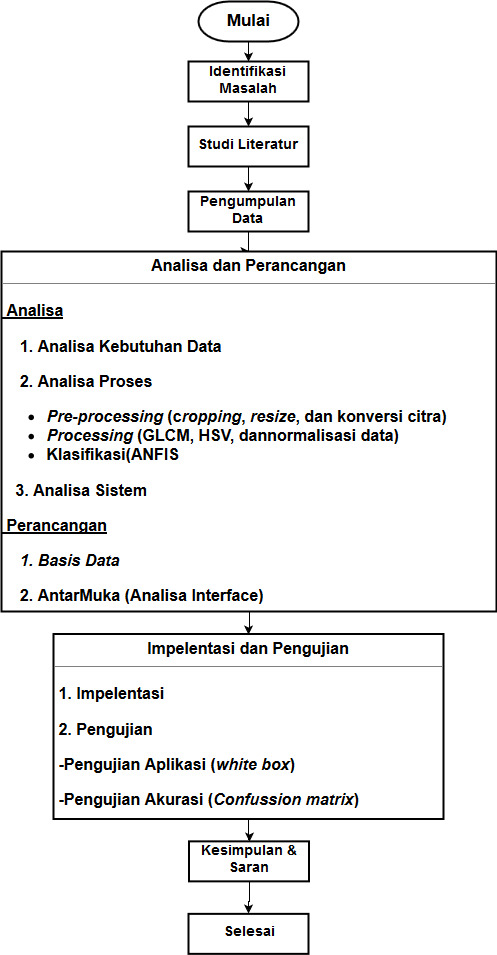
Tabel 2.1 Penelitian Terkait.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Penulis | Judul | Metode | Hasil |
| 1 | Permana (2017) | klasifikasi jenis-jenis buah jeruk dengan metode *K-Nearest Neighbor* | *K-Nearest Neighbor* | Dari 4 data skenario uji coba, didapatlah akurasi sebesar 63%, 55%, 54%, dan 58%. |
| 2 | Fuzzy Yustika, Kana Saputra (2017) | klasifikasi belimbing menggunakan metode *Naïve Bayes,* dengan ekstraksi fitur nilai ­*Red,Green, Blue* | *Naïve Bayes,* dengan ekstraksi fitur nilai ­*Red,Green, Blue* | . Hasil klasifikasi menunjukkan hasil akurasi sebesar 80%. |
| 3 | Mega (2018) | aplikasi pengolahan citra untuk klasifikasi buah manga | *Learining Vector Quantization* (LVQ) untuk mendapatkan ciri pada setiap karakter masukan buah manga dalam perhitungan nilai RGB | Berdasarkan hasil uji coba pada aplikasi menghasilkan akurasi sebesar 86.6%. |
| 4 | Restu Widodo, Agus Wahyu (2018) | *Grey Level Co-occurrence Matrix* (GLCM) pada citra buah jeruk keprok untuk menentukan klasifikasi mutu | *Grey Level Co-occurrence Matrix* (GLCM), *support Vector Machine* (SVM) | Penelitian ini menghasilkan tingkat akurasi terbaik sebesar 82,5%. |
| 5 | Elvia Budianita, Jasril, & Handayani ( 2015) | Implementasi Citra dan Klasifikasi K*-Nearest Neighbour* untuk membangun Aplikasi Pembeda Daging Sapi dan Daging Babi | K*-Nearest Neighbour,* metode HSV | tingkat keakurasian 88,75% |
| 6 | Retno Nugroho Whidhiasih,  Nursinta  Supriyanto A.W (2012) | Identifikasi Buah Belimbing berdasarkan Citra *Red-Green-Blue* Menggunakan *Adaptive Neuro Fuzzy Inference System* (Anfis) | *Adaptive Neuro Fuzzy Inference System* (ANFIS)  Berdasarkan citra RGB (*red, green, blue*) | Akurasi yang diperoleh yakni 100% untuk kelas asam, 100% untuk kelas sedang, dan 67% untuk kelas manis. |
| 7 | Nur Sakinah, Retno Nugroho (2017) | mengidentifikasi buah manga gedong gincu Cirebon dengan metode *Adaptive Neuro Fuzzy Interference System* (ANFIS) | *Adaptive Neuro Fuzzy Interference System* (ANFIS), dengan metode ekstraksi *Red-Green-Blue* (RGB) | Pada pelatihan ANFIS menggunakan algoritma *hybrid* dengan metode  *Recursive Least Square Estimator*  (RLSE), dan jumlah *Epoch* sebanyak 1000, nilai error 0.5220 dan mampu mengidentifikasi jenis buah Mangga Gedong Gincu Cirebon dengan tingkat akurasi sebesar 66,6% |
| 8. | Muhammad Asyhar Agmalaro,  Aziz Kustiyo,  Auriza Rahmad Akbar (2013) | Identifikasi Tanaman Buah Tropika Berdasarkan Tekstur Permukaan Daun Menggunakan Jaringan Syaraf Tiruan | JST *backpropagation* berdasarkan ekstraksi fitur *grey level co-occurrence matrix* (GLCM) | Identifikasi yang dilakukan pada tanaman buah tropika ini menggunakan 15 sampel citra daun buah-buahan tropika. Citra daun yang telah di proses dengan ekstraksi GLCM akan dijadikan *input* untuk proses indentifikasi. *Hidden neuron* yang digunakan pada pengujian ini sebanyak 7 lapis dan memperoleh hasil akurasi yaitu 90%. |
| 9. | Anita Ahmad Kasim,  Agus Harjoko (2014) | Klasifikasi Citra Batik Menggunakan Jaringan Syaraf Tiruan Berdasarkan *Gray Level Co-Occurrence Matrices* (GLCM) | *JST backpropagation dan GLCM* | Pada klasifikasi citra batik ini diterapkan jaringan saraf tiruan menggunakan algoritma *backpropagation*. Hasil akurasi pada data latih sebesar 95,7% , daru 37 (91.9%) citra baik non geometri terdapat 3 (81%) ciri yang tidak dapat diklasifikasikan kedalam kelasnya. Sedangkan ciri citra geometri dapat diklasifikasikan dengan benar 100%. |
| 10 | Maura Widyaningsih (2016) | Identifikasi Kematangan Buah Apel Dengan *Gray Level Co-Occurrence Matrix* (Glcm) | *GLCM dengan faktor eucledian distance* | Identifikasi kematangan buah apel dilakukan dengan 12 data latih citra apel. 12 data latih tersebut dibagi menjadi 3 kategori yaitu mentah, setengah matang, dan matang. Pengujian GLCM dengan sudut 0 derajat hasil ekstraksi ciri citra uji dapat dikenali dengan faktor *eucledian distance* terhadap citra *queri.* |

**BAB III**

**METODOLOGI PENELITIAN**

Metode penelitian merupakan tahapan yang menjadi acuan suatu penelitian untuk mencapai tujuan dari penelitian. Berikut Gambar 3.1 adalah tahapan-tahapan penting yang akan dilakukan pada penelitian ini.



**Gambar 3.1 Metodologi Penelitian**

1. **Identifikasi Masalah**

Tahap identifikasi masalah adalah tahap dimana peneliti melakukan identifikasi terhadap masalah yang ada, yang dimana pada kasus ini yang di ambil adalah pohon gaharu. Kurangnya ilmu pengetahuan tentang gaharu membuah hal ini menjadi salah satu masalah utamanya. Pengetahuan yang dimaksud disini adalah pengetahuan mendalam tentang pohon gaharu, baik jenisnya maupun pengolahannya. Para petani gaharu dan mahasiswa yang studi dibidang kehutan dan pertanianpun juga tidak semuanya mengetahui jenis-jenis pohon gaharu. Pohon gaharu memiliki nilai jual serta nilai sosial yang tinggi, sehingga pengetahuan tentang gaharu perlu untuk diketahui dan dipelajari. Perbedaan pohon gaharu dapat dilihat dari 3 aspek yaitu, daun, batang dan buah. Buah pohon gaharu menjadi salah satu ciri fisik untuk mengetahui jenis dari pohon gaharu. Untuk melakukan identifikasi jenis gaharu 1 buah pohon Gaharu, buah pohon haru dibelah dua terlebih dahulu, setelah dibelah isi bagian dalam buah akan terlihat sama oleh mata awam. Dari hal tersebut maka peneliti merumuskan masalah untuk melakukan penelitian tugas akhir dengan judul **“Klasifikasi jenis pohon gaharu berdasarkan tekstur,bentuk dan warna pada buah menggunakan metode *adaptive neuro fuzzy inference system* (anfis)”.**

1. **Studi Literatur**

Tahap awal yang dilakukan pada penelitian ini adalah studi pustaka. Pada penelitian ini, peneliti mengumpulkan informasi terkait dengan pengolahan citra digital menggunakan ekstraksi fitur tekstur *Gray Level Co-occurrence Matrix* (GLCM), fitur warna *Hue Saturation Value* (HSV), metode *Adaptive Neural Fuzzy Inference System* (ANFIS), dan pohon gaharu. Pengumpulan informasi dilakukan melalui buku dan jurnal penelitian.

1. **Pengumpulan Data**

Data yang digunakan pada penelitian kali ini adalah data primer berupa buah pohon gaharu. Buah pohon gaharu didapatkan dari pohon gaharu yang dibudidayakan oleh GAHARU PLAZA INDONESIA yang beralamat Jl. Guru Blok A No.12 Arengka Raya Residence Sidumulyo Pekanbaru. Pohon gaharu yang dibudidayakan oleh gaharu plaza indonesia terdapat di daerah kualu nenas daerah kampar. Pengumpulan data dilakukan dengan cara mengambil 120 citra buah dari 6 jenis pohon gaharu. Alat yang digunakan untuk pengambilan citra buah pohon gaharu adalah kamera DSLR canon EOS 60D. Data citra yang akan digunakan pada penelitian ini adalah data citra yang berformat .jpg. pengambilan citra buah dilakukan menggunakan *background* berwarna putih serta buah yang akan diambil citrannya adalah bagian dalam buah, sehingga buah harus di belah terlebih daulu supaya dapat memberikan hasil citra dengan tekstur yang jelas.

1. **Analisa dan Perancangan**

Pada tahap analisa ini dilakukan penelusuran kebutuhan yang diperlukan dalam membangun aplikasi ini. Analisa dilakukan terhadap data yang telah diperoleh. Pada penelitian ini terdadapat analisa kebutuhan data dan analisa proses.

1. **Analisa Kebutuhan Data**

Tahapan analisa kebutuhan data dilakukan untuk menganalisa data yang dibutuhkan dalam penelitian. Pada tahap ini akan ditentukan kriteria data yang dapat digunakan pada penelitian sesuai dengan data yang telah dikumpulkan.

1. **Analisa Proses**

Tahapan analisa proses terdiri dari tiga tahap yaitu *pre-processing*, *processing*, dan klasifikasi. Tahapan analisa proses dilakukan untuk mengantisipasi kesalahan yang terjadi ketika implementasi, dikarenakan tidak cocoknya data yang dimiliki dengan data yang dibutuhkan. Berikut penjelasan 3 proses pada analisa proses.

1. ***Pre-processing***

Tahap *pre-processing* dilakukan untuk mendapatkan citra baru yang akan digunakan untuk proses ekstraksi fitur. Ada 3 tahapan yang dilakukan pada saat *pre-processing* ini.

1. *Cropping*

Data citra buah yang dikumpulkan akan dilakukan perbaikan citra dengan cara melakukan proses *Cropping*. *Cropping* dilakukan dengan memotong latar belakang atau *background* pada buah yang menghasilkan citra yang berfokus pada buah pohon gaharu dengan perbandingan rationya 2:3.

1. *Resize*

Tahap ini mengubah ukuran citra pada buah. Tahap ini dilakukan agar menghasilkan ukuran citra buah gaharu yang sama. Citra yang telah di *resize* disamakan ukurannya menjadi 200x300 piksel.

1. Konversi Citra

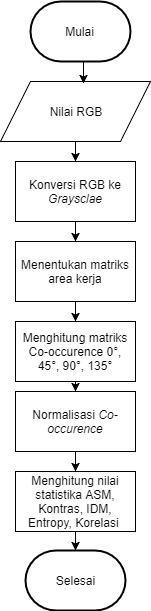
Citra buah yang telah *cropping* dan *resize* akan dilakukan konversi citra. Citra buah gaharu yang awalnya berwarna (RGB) akan dikonversikan menjadi citra keabuan (*grayscale*) dengan persamaan (2:3) untuk digunakan pada ekstraksi fitur GLCM.

1. ***Processing***

Citra buah gaharu yang telah diolah pada tahap *pre-processing* akandiolah kembali untuk mendapatkan nilai ekstrasi warna *Hue Saturation Value* (HSV) serta fitur tekstur GLCM pada buah. Berikut tahapan yang dilakukan pada tahapan *processing* ini.

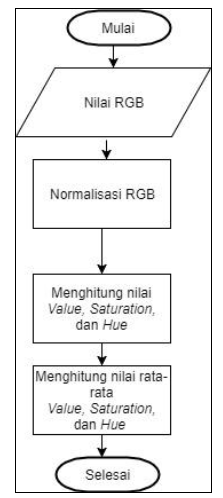
1. Ekstraksi Fitur Tekstur *Gray Level Co-occurence Matrix*  (GLCM)

Nilai tekstur buah didapat dengan melakukan ekstraksi fitur tekstur. Fitur tekstur yang digunakan adalah *Gray Level Co-occurence Matrix* (GLCM). Citra *grayscale* yang telah didapat akan dibuat sebuah matriks kookurensi dengan ukuran 256 × 256. Matriks tersebut dihitung dengan jarak 1 yang berdasarkan sudut 0º, 45º, 90º, 135º sehingga menghasilkan 4 matriks kookurensi. Hasil dari keempat matriks kookurensi akan dirata-ratakan dan kemudian digunakan untuk mendapatkan fitur ciri tekstur yang digunakan. Fitur ciri tekstur yang digunakan ada 5, yaitu *angular second moment* (ASM), kontras, inverse different moment, entropi dan korelasi. Berikut merupakan Alur proses dari GLCM:

**Gambar 3.2 Alur Proses GLCM**

1. Ekstraksi Fitur Tekstur *Hue Saturation Value* (HSV)

Tahapan ini dilakukan untuk ekstraksi ciri warna pada data citra setelah dilakukannya proses *cropping.* Proses pada tahap *processing* menggunakan metode HSV. proses dimulai dengan memasukkan data citra buah yang telah di melakukan proses pre-*processing* supaya mendapat citra yang diinginkan serta untuk dapat mempercepat proses komputasi, setelah didapat proses *cropping* maka dilakukan proses ekstraksi ciri warna dimana proses ini dimulai dengan mencari nilai RGB *(red, green, blue)* untuk dikonversikan ke HSV.



**Gambar 3.3 Alur Proses HSV**

1. Normalisasi Data

Setelah didapatkan nilai ekstraksi fitur tekstur GLCM dan ekstraksi fitur warna HSV, akan dilakukan normalisasi data untuk mengubah nilai menjadi rentang 0 sampai 1. Normalisasi yang dihitung akan mendapatkan hasil untuk digunakan sebagai *input* pada metode klasifikasi ANFIS.

1. **Klasifikasi**

Pada tahap klasifikasi pada buah gaharu ini dilakukan menggunakan metode *Adaptive Neuro Fuzzy Inference System* (ANFIS). Pada proses pelatihan ANFIS, model inferensi yang digunakan adalah model Sugeno orde nol dimana ouput sistem berupa konstanta. Sistem inferensi *fuzzy* merupakan proses pemetaan dari suatu *input* ke *ouput* dengan menggunakan logika *fuzzy.*

1. **Analisa Sistem**

Tahap analisa sistem ini merupakan tahap dimana alur dari sistem digambarkan dalam sebuah rancangan sistem yang akan dijadikan acuan pada sistem yang akan dibangun. Adapun perangkat analisa sistem yang digunakan adalah :

### *Usecase* *Diagram*

*use case* diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Shalahuddin M., 2014)*.*

### *Activity Diagram*

Diagram aktivitas atau *Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses atau meu yang ada pada perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Shalahuddin M., 2014).

### *Sequence Diagram*

*Sequence Diagram* menggambarkan *action* dari objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *massage* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *usecase* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu (Shalahuddin M., 2014).

### *Class Diagram*

*lCass diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan *method* atau operasi. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas dan Operasi atau *method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas (Shalahuddin M., 2014).

1. **Perancangan**

Tahap perancangan ini adalah suatu proses yang dilakukan dalam merancang alur sistem. Tahap ini menjadi acuan untuk tahap implementasi berikutnya. Alur sistem yang dibuat pada penelitian ini adalah *Flowchart* (diagram alur) dan perancangan *User Interface* (antar muka) aplikasi. Diagram alur ini berguna untuk mengambarkan alur proses dari sistem yang akan dibangun, sedangkan *User Interface* dirancang ini berguna sebagai acuan dalam membuat aplikasi atau sistem.

1. **Implementasi dan pengujian**

Tahap implementasi dan pengujian ini adalah tahapan yang dilakukan pada sistem berdasarkan tahapan analisa dan perancangan sebelumnya.

1. **Implementasi**

Pada tahap implementasi ini adalah tahap penerapan rancangan yang sudah dibuat sebelumnya menjadi sebuah aplikasi/sistem yang utuh. Adapun lingkungan dari implementasi yang terdiri dari lingkungan perangkat keras (*hardware*) dan perangkat lunak (*software*). Berikut alat pendukung pada lingkungan implementasi yang dibutuhkan untuk membangun sistem.

1. Perangkat Keras (*Hardware*)

*Processor* : Intel® Core(TM) i5-4200 CPU 1.60 GHz

(CPUs)

*Memory* : 4096MB RAM

*Hard Disk* : 500 GB

1. Perangkat Lunak (*Software*)

Sistem Operasi : Windows 7 Ultimate 64-bit (6.1, Build 7061)

Web Server : Apache

*Browser* : Google Chrome Version  74.0.3729.131 (Official Build) 64-bit

Bahasa Pemograman : PHP, HTML, Javascript

Tool : Sublime Text 3

DBMS : MySQL

1. **Pengujian**

Tahap pengujian ini dilakukan untuk mendapatkan tujuan yang diinginkan agar berjalan dengan baik. Hasil yang diharapkan pada pengujian ini adalah sistem dapat mengenali ciri buah secara jelas sehingga dapat mengetahui jenis pohon gaharu. Terdapat dua jenis pengujian yang akan dilakukan yaitu pengujian pada *source code* sistem dan pengujian akurasi menggunakan *confusion matrix* dengan pembagian datanya menggunakan *k-fold cross validation*.

Pengujian dilakukan dengan *white box testing* melakukan pengecekan kepada seluruh elemen program sistem seperti perulangan (*loop*), logika keputusan dan alur logika sistem. Pengujian ini bertujuan untuk mengetahui program berjalan baik secara tepat. Pengujian ini dilakukan berdasarkan kode program. Selanjutnya pengujian akurasi *confusion matrix* dilakukan dengan menghubungkan hasil klasifikasi atau pemetaan dengan hasil data yang diperoleh untuk uji akurasi. Pengujian akurasi ini melakukan pembagian datanya menggunakan *k-fold cross validation.* Pembagian seluruh data citra gaharu akan dibagi menjadi 10 subset, yaitu *fold* 1, *fold* 2, *fold* 3, *fold* 4, *fold* 5, *fold* 6,  *fold* 7, *fold* 8, *fold* 9, *fold* 10. Hal tersebut dilakukan bertujuan agar memperoleh hasil terbaik. Pelatihan akan dilakukan perulangan sebanyak 10 kali. Pada setiap perulangan 1 *fold* akan dijadikan data uji dan 9 *fold* lainnya akan dijadikan data latih. Proses ini dilakukan sampai semua *fold* pernah menjadi sebagai data latih dan data uji.

1. **Kesimpulan dan saran**

Tahap kesimpulan dan saran ini adalah tahap akhir dari penelitian. Kesimpulan berisi hasil dari penelitian yang telah dilakukan sedangkan saran berisi hal-hal yang disarakan peneliti agar dapat dilakukan pada penelitian yang dilakukan selanjutnya dengan hasil yang lebih baik.

**BAB IV**

**ANALISA DAN PERANCANGAN**

## **Analisa**

Analisa merupakan suatu proses untuk mengetahui permasalahan yang terjadi pada penelitian. Analisan ini bertujuan agar dapat memberikan solusi dan pemahan terhadap permasalahan yang diuraikan pada penelitian. Pada penelitian ini akan dilakukan analisa kebutuhan data dan proses penelitian.

### Analisa Kebutuhan Data

Analisa kebutuhan data ini dilakukan dengan menganalisa data yang diperlukan pada penelitian. Pada penelitian kali ini citra yang dipakai adalah citra buah pohon gaharu. Buah yang digunakan untuk pengambilan citra pada penelitian kali ini memiliki beberapa standard:

* + 1. Citra buah yang digunakan hanya bagian luar buah.
    2. Citra buah diambil dengan latar belakang buah berwarna putih.
    3. Citra buah diambil dengan jarak 20 cm.
    4. Citra daun diambil menggunakan kamera DSLR EOS 60D dengan ISO-speed 400 dengan kondisi *flash* kamera dinyalakan.
    5. Citra daun yang digunakan memiliki format .jpg.

### Analisa Proses

Analisa proses yang akan dilakukan terbagi menjadi tiga tahapan yaitu *pre-processing*, *processing*, dan klasifikasi. Tahap *pre-processing* ini adalah tahapan pengolahan yang dilakukan pada citra daun sebelum citra di ekstraksi. Tahap *processing* dilakukan ketika citra daun sudah di *crop, resize,* dan konversi citrapada tahap sebelumnya. Tahap *processing* ini melakukan perhitungan terhadap ekstaksi ciri daun yang terdiri dari ekstraksi fitur morfologi digital, ekstraksi fitur warna HSV, dan normalisasi data. Tahap berikutnya adalah klasifikasi citra daun menggunakan ANFIS.



Gambar 4.1 Analisa Proses Klasifikasi Daun Gaharu

#### Pre-Processing

Tahap *pre-processing* ini dilakukan untuk mendapatkan data citra yang digunakan pada proses ekstraksi citra nantinya. Citra daun awal yang digunakan memiliki ukuran 3456 x 5184 px dengan format .jpg. Berikut gambar 4.2 Citra Awal Daun Gaharu *Sinensis*.

v

Gambar 4.2 Citra Awal Daun Buah *Sinensis*

* 1. ***Cropping***

Tahap *cropping* adalah tahapan dimana dilakukan teknik pengolahan citra dengan memotong citra pada ukuran tertentu. Pada penelitian ini citra buah gaharu akan di *cropping* dengan ukuran kanvas 2x3 cm.



Gambar 4.3 Citra Buah Gaharu *Sinensis* Setelah *Cropping*

Citra buah yang telah di *crop* akan merubah ukuran piksel dari citra tersebut sehingga ukuran citra buah menjadi 600x900px. Tahap *cropping*  ini dilakukan agar citra yang diambil bisa terfokus pada citra buah.

* 1. ***Resize***

Tahap *resize* adalah tahapan yang dilakukan dengan merubah ukuran piksel pada citra buah gaharu dengan rasio 2:3.

Gambar 4.4 Citra Buah Gaharu *Sinensis* Setelah di *Resize*

Ukuran piksel citra buah gaharu *sinensis* yang telah di *resize* berubah menjadi 200x300 px. Tujuan dari tahapan *resize* ini agar mempercepat proses konversi citra yang dilakukan pada tahap selanjutnya.

* 1. **Konversi Citra**

Tahap konversi citra adalah proses perubahan dari suatu citra menjadi citra lainnya. Pada penelitian ini konversi citra yang dilakukan yaitu konversi citra berwarna (*Red, Green, Blue*) menjadi citra *grayscale* dan selanjutnya citra *grayscale* akan dikonversikan menjadi citra biner. Berikut Tabel 4.1 Nilai Piksel *Red*, Tabel 4.2 Nilai Piksel *Green* dan Tabel 4.3 Nilai Piksel *Blue*.

Tabel 4.1 Nilai Piksel *Red*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **...** | **49** | **50** | **...** | **99** | **100** | **...** | **149** | **150** | **...** | **199** | **200** |
| **1** | 196 | 196 | ... | 199 | 199 | ... | 199 | 199 | ... | 199 | 200 | ... | 204 | 205 |
| **2** | 196 | 196 | ... | 199 | 199 | ... | 199 | 199 | ... | 199 | 201 | ... | 204 | 205 |
| **3** | 196 | 197 | ... | 200 | 200 | ... | 199 | 199 | ... | 199 | 201 | ... | 203 | 203 |
|  |  |  | ... |  |  | ... |  |  | ... |  |  | ... |  |  |
| **98** | 203 | 203 | ... | 205 | 205 | ... | 15 | 18 | ... | 201 | 201 | ... | 208 | 207 |
| **99** | 203 | 203 | ... | 205 | 205 | ... | 18 | 27 | ... | 201 | 201 | ... | 207 | 207 |
| **100** | 203 | 203 | ... | 205 | 205 | ... | 27 | 35 | ... | 201 | 201 | ... | 205 | 206 |
|  |  |  | ... |  |  | ... |  |  | ... |  |  | ... |  |  |
| **198** | 204 | 204 | ... | 203 | 203 | ... | 118 | 115 | ... | 200 | 200 | ... | 201 | 201 |
| **199** | 204 | 204 | ... | 203 | 203 | ... | 112 | 114 | ... | 200 | 200 | ... | 201 | 201 |
| **200** | 204 | 204 | ... | 203 | 203 | ... | 110 | 116 | ... | 201 | 200 | ... | 201 | 201 |
|  |  |  | ... |  |  | ... |  |  | ... |  |  | ... |  |  |
| **298** | 199 | 199 | ... | 200 | 200 | ... | 201 | 201 | ... | 199 | 198 | ... | 195 | 195 |
| **299** | 198 | 199 | ... | 200 | 200 | ... | 201 | 201 | ... | 199 | 198 | ... | 195 | 195 |
| **300** | 198 | 199 | ... | 200 | 200 | ... | 201 | 201 | ... | 199 | 198 | ... | 195 | 195 |

Tabel 4.2 Nilai Piksel *Green*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **...** | **49** | **50** | **...** | **99** | **100** | **...** | **149** | **150** | **...** | **199** | **200** |
| **1** | 206 | 206 | ... | 209 | 209 | ... | 209 | 209 | ... | 209 | 209 | ... | 212 | 213 |
| **2** | 206 | 206 | ... | 209 | 209 | ... | 209 | 209 | ... | 209 | 209 | ... | 212 | 213 |
| **3** | 206 | 207 | ... | 210 | 210 | ... | 209 | 209 | ... | 209 | 210 | ... | 211 | 211 |
|  |  |  | ... |  |  | ... |  |  | ... |  |  | ... |  |  |
| **98** | 212 | 212 | ... | 215 | 215 | ... | 47 | 51 | ... | 211 | 211 | ... | 215 | 214 |
| **99** | 212 | 212 | ... | 215 | 215 | ... | 49 | 57 | ... | 211 | 211 | ... | 214 | 214 |
| **100** | 212 | 212 | ... | 215 | 215 | ... | 54 | 63 | ... | 211 | 211 | ... | 212 | 213 |
|  |  |  | ... |  |  | ... |  |  | ... |  |  | ... |  |  |
| **198** | 214 | 214 | ... | 213 | 213 | ... | 142 | 139 | ... | 211 | 211 | ... | 209 | 209 |
| **199** | 214 | 214 | ... | 213 | 213 | ... | 135 | 137 | ... | 211 | 211 | ... | 209 | 209 |
| **200** | 214 | 214 | ... | 213 | 213 | ... | 131 | 136 | ... | 212 | 211 | ... | 209 | 209 |
|  |  |  | ... |  |  | ... |  |  | ... |  |  | ... |  |  |
| **298** | 209 | 209 | ... | 210 | 210 | ... | 209 | 209 | ... | 207 | 206 | ... | 200 | 200 |
| **299** | 208 | 209 | ... | 210 | 210 | ... | 209 | 209 | ... | 207 | 206 | ... | 200 | 200 |
| **300** | 208 | 209 | ... | 210 | 210 | ... | 209 | 209 | ... | 207 | 206 | ... | 200 | 200 |

Tabel 4.3 Nilai Piksel *Blue*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **...** | **49** | **50** | **...** | **99** | **100** | **...** | **149** | **150** | **...** | **199** | **200** |
| **1** | 210 | 210 | ... | 213 | 213 | ... | 213 | 213 | ... | 212 | 211 | ... | 214 | 215 |
| **2** | 210 | 210 | ... | 213 | 213 | ... | 213 | 213 | ... | 212 | 211 | ... | 214 | 215 |
| **3** | 210 | 211 | ... | 214 | 214 | ... | 213 | 213 | ... | 212 | 212 | ... | 213 | 213 |
|  |  |  | ... |  |  | ... |  |  | ... |  |  | ... |  |  |
| **98** | 220 | 220 | ... | 218 | 218 | ... | 18 | 23 | ... | 215 | 215 | ... | 219 | 218 |
| **99** | 220 | 220 | ... | 218 | 218 | ... | 18 | 27 | ... | 215 | 215 | ... | 218 | 218 |
| **100** | 220 | 220 | ... | 218 | 218 | ... | 21 | 29 | ... | 215 | 215 | ... | 217 | 218 |
|  |  |  | ... |  |  | ... |  |  | ... |  |  | ... |  |  |
| **198** | 217 | 217 | ... | 216 | 216 | ... | 88 | 84 | ... | 213 | 213 | ... | 211 | 211 |
| **199** | 217 | 217 | ... | 216 | 216 | ... | 82 | 84 | ... | 213 | 213 | ... | 211 | 211 |
| **200** | 217 | 217 | ... | 216 | 216 | ... | 80 | 85 | ... | 214 | 213 | ... | 211 | 211 |
|  |  |  | ... |  |  | ... |  |  | ... |  |  | ... |  |  |
| **298** | 213 | 213 | ... | 214 | 214 | ... | 211 | 211 | ... | 209 | 208 | ... | 203 | 203 |
| **299** | 212 | 213 | ... | 214 | 214 | ... | 211 | 211 | ... | 209 | 208 | ... | 203 | 203 |
| **300** | 212 | 213 | ... | 214 | 214 | ... | 211 | 211 | ... | 209 | 208 | ... | 203 | 203 |

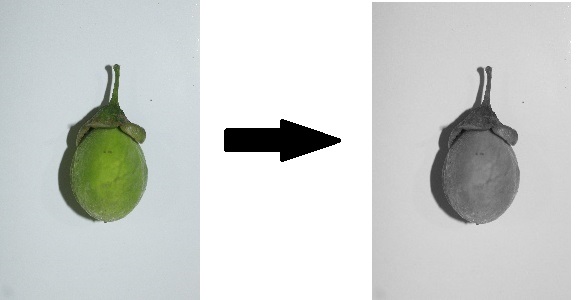
Berdasarkan nilai piksel dari citra RGB diatas, maka nilai piksel citra *grayscale* yang didapat menggunakan Persamaan (2.37). contoh perhitungan pada piksel (70,70) dilakukan dengan cara sebagai berikut.

Berdasarkan perhitungan diatas dihasilkanlah citra *grayscale* yang dapat dilihat pada Tabel 4.4 sebagai berikut.

Tabel 4.4 Nilai Piksel *Grayscale*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **...** | **49** | **50** | **...** | **99** | **100** | **...** | **149** | **150** | **...** | **199** | **200** |
| **1** | 203 | 203 | ... | 206 | 206 | ... | 206 | 206 | ... | 207 | 207 | ... | 210 | 211 |
| **2** | 203 | 203 | ... | 206 | 206 | ... | 206 | 206 | ... | 207 | 207 | ... | 210 | 211 |
| **3** | 203 | 204 | ... | 207 | 207 | ... | 206 | 206 | ... | 208 | 208 | ... | 209 | 209 |
|  |  |  | ... |  |  | ... |  |  | ... |  |  | ... |  |  |
| **98** | 210 | 210 | ... | 212 | 212 | ... | 34 | 38 | ... | 208 | 208 | ... | 213 | 212 |
| **99** | 210 | 210 | ... | 212 | 212 | ... | 36 | 45 | ... | 208 | 208 | ... | 212 | 212 |
| **100** | 210 | 210 | ... | 212 | 212 | ... | 42 | 51 | ... | 208 | 208 | ... | 210 | 211 |
|  |  |  | ... |  |  | ... |  |  | ... |  |  | ... |  |  |
| **198** | 211 | 211 | ... | 210 | 210 | ... | 129 | 126 | ... | 208 | 208 | ... | 207 | 207 |
| **199** | 211 | 211 | ... | 210 | 210 | ... | 122 | 124 | ... | 208 | 208 | ... | 207 | 207 |
| **200** | 211 | 211 | ... | 210 | 210 | ... | 119 | 124 | ... | 208 | 208 | ... | 207 | 207 |
|  |  |  | ... |  |  | ... |  |  | ... |  |  | ... |  |  |
| **298** | 206 | 206 | ... | 207 | 207 | ... | 207 | 207 | ... | 204 | 204 | ... | 199 | 199 |
| **299** | 205 | 206 | ... | 207 | 207 | ... | 207 | 207 | ... | 204 | 204 | ... | 199 | 199 |
| **300** | 205 | 206 | ... | 207 | 207 | ... | 207 | 207 | ... | 204 | 204 | ... | 199 | 199 |

Berdasarkan matrik citra *grayscale* diatas maka citra yang sebelumnya bewarna RGB akan diubah menjadi citra *grayscale* seperti yang terlihat pada Gambar 4.5 berikut.



Gambar 4.5 Transformasi Citra RGB ke Citra *Grayscale*

Selanjutnya citra *grayscale* tersebut akan dikonversikan menjadi citra biner menggunakan Persamaan (2.38). Pada penelitian ini konversi citra *grayscale* menjadi citra biner menggunakan nilai *threshold* (*a*) = 170 karena nilai intensitas citra yang lebih atau sama dengan *threshold* akan diubah menjadi nilai 1 (warna putih) sedangkan nilai yang kurang dari *threshold* akan diubah menjadi nilai 0 (warna hitam). Berikut perhitungan konversi citra *grayscale* menjadi citra biner.

Berdasarkan perhitungan di atas, maka nilai piksel citra biner yang didapat menggunakan Persamaan (2.38) dapat dilihat pada Tabel 4.5 sebagai berikut.

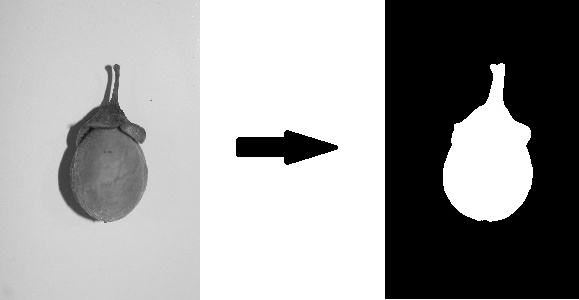
Tabel 4.5 Nilai Piksel Citra Biner

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **...** | **49** | **50** | **...** | **99** | **100** | **...** | **149** | **150** | **...** | **199** | **200** |
| **1** | 1 | 1 | ... | 1 | 1 | ... | 1 | 1 | ... | 1 | 1 | ... | 1 | 1 |
| **2** | 1 | 1 | ... | 1 | 1 | ... | 1 | 1 | ... | 1 | 1 | ... | 1 | 1 |
| **3** | 1 | 1 | ... | 1 | 1 | ... | 1 | 1 | ... | 1 | 1 | ... | 1 | 1 |
|  |  |  | ... |  |  | ... |  |  | ... |  |  | ... |  |  |
| **98** | 1 | 1 | ... | 1 | 1 | ... | 0 | 0 | ... | 1 | 1 | ... | 1 | 1 |
| **99** | 1 | 1 | ... | 1 | 1 | ... | 0 | 0 | ... | 1 | 1 | ... | 1 | 1 |
| **100** | 1 | 1 | ... | 1 | 1 | ... | 0 | 0 | ... | 1 | 1 | ... | 1 | 1 |
|  |  |  | ... |  |  | ... |  |  | ... |  |  | ... |  |  |
| **198** | 1 | 1 | ... | 1 | 1 | ... | 0 | 0 | ... | 1 | 1 | ... | 1 | 1 |
| **199** | 1 | 1 | ... | 1 | 1 | ... | 0 | 0 | ... | 1 | 1 | ... | 1 | 1 |
| **200** | 1 | 1 | ... | 1 | 1 | ... | 0 | 0 | ... | 1 | 1 | ... | 1 | 1 |
|  |  |  | ... |  |  | ... |  |  | ... |  |  | ... |  |  |
| **298** | 1 | 1 | ... | 1 | 1 | ... | 1 | 1 | ... | 1 | 1 | ... | 1 | 1 |
| **299** | 1 | 1 | ... | 1 | 1 | ... | 1 | 1 | ... | 1 | 1 | ... | 1 | 1 |
| **300** | 1 | 1 | ... | 1 | 1 | ... | 1 | 1 | ... | 1 | 1 | ... | 1 | 1 |

Selanjutnya hasil citra biner diinvers sehingga menghasilkan Tabel 4.6

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **...** | **49** | **50** | **...** | **99** | **100** | **...** | **149** | **150** | **...** | **199** | **200** |
| **1** | 0 | 0 | ... | 0 | 0 | ... | 0 | 0 | ... | 0 | 0 | ... | 0 | 0 |
| **2** | 0 | 0 | ... | 0 | 0 | ... | 0 | 0 | ... | 0 | 0 | ... | 0 | 0 |
| **3** | 0 | 0 | ... | 0 | 0 | ... | 0 | 0 | ... | 0 | 0 | ... | 0 | 0 |
|  |  |  | ... |  |  | ... |  |  | ... |  |  | ... |  |  |
| **98** | 0 | 0 | ... | 0 | 0 | ... | 1 | 1 | ... | 0 | 0 | ... | 0 | 0 |
| **99** | 0 | 0 | ... | 0 | 0 | ... | 1 | 1 | ... | 0 | 0 | ... | 0 | 0 |
| **100** | 0 | 0 | ... | 0 | 0 | ... | 1 | 1 | ... | 0 | 0 | ... | 0 | 0 |
|  |  |  | ... |  |  | ... |  |  | ... |  |  | ... |  |  |
| **198** | 0 | 0 | ... | 0 | 0 | ... | 1 | 1 | ... | 0 | 0 | ... | 0 | 0 |
| **199** | 0 | 0 | ... | 0 | 0 | ... | 1 | 1 | ... | 0 | 0 | ... | 0 | 0 |
| **200** | 0 | 0 | ... | 0 | 0 | ... | 1 | 1 | ... | 0 | 0 | ... | 0 | 0 |
|  |  |  | ... |  |  | ... |  |  | ... |  |  | ... |  |  |
| **298** | 0 | 0 | ... | 0 | 0 | ... | 0 | 0 | ... | 0 | 0 | ... | 0 | 0 |
| **299** | 0 | 0 | ... | 0 | 0 | ... | 0 | 0 | ... | 0 | 0 | ... | 0 | 0 |
| **300** | 0 | 0 | ... | 0 | 0 | ... | 0 | 0 | ... | 0 | 0 | ... | 0 | 0 |

Berikut Gambar 4.6 Hasil konversi citra RGB menjadi citra *grayscale* dan dikonversikan kembali menjadi citra biner lalu di inver.



Gambar 4.6 Transformasi Citra *Grayscale* ke Citra Biner Invers

#### Processing

Tahap *processing* pada penelitian ini menerapkan ekstraksi fitur teksur GLCM dan fitur morfologi digital untuk mendapatkan nilai ekstraksi ciri daun.

1. **Ekstraksi Fitur Morfologi Digital**

Fitur morfologi digital yang digunakan pada penelitian ini berfungsi untuk menghitung bentuk fisik dari citra daun gaharu. Fitur ini dihitung dengan masukannya yaitu citra daun gaharu yang sudah dikonversikan ke citra biner. Terdapat 2 ciri pada fitur morfologi digital yaitu ciri dasar dan ciri turunan. Perhitungan pada fitur ini menggunakan 2 tahapan, pertama menghitung ciri dasar terlebih dahulu dan kedua menghitung ciri turunan. Berikut Gambar 4.11 *Flowchart* Fitur Morfologi Digital.



**Gambar 4.11 *Flowchart* Fitur Morfologi Digital**

1. Citra Biner

Pada perhitungan ekstraksi fitur morfologi digital ini membutuhkan sebuah masukan atau *input* yang diperlukan. Data citra binerpada Gambar 4.5 yang sudah didapat dalam tahap *pre-processing* menjadi *input* pada perhitungan tersebut.

1. Menghitung Ciri Dasar

Ciri dasar pada fitur morfologi digital ini adalah nilai yang didapat dari bentuk daun. Terdapat 5 ciri dasar pada perhitungan ini yaitu diameter, *physiological length, physiological width,* area dan perimeter. Nilai ciri dasar ini digunakan untuk menghitung nilai ciri turunan. Berdasarkan Tabel 4.5 nilai piksel citra biner yang telah didapat, maka berikut Tabel 4.15 nilai ciri dasar fitur morfologi.

**Tabel 4.15 Nilai Ciri Dasar Fitur Morfologi**

|  |  |
| --- | --- |
| Ciri Dasar Fitur Morfologi | Nilai Piksel |
| Diameter (D) | 105,1937 |
| *Physiological of Length* (Lp) | 138,5485 |
| *Physiological of Width* (Wp) | 85,9732 |
| Area (A) | 86911 |
| Perimeter (P) | 412,7740 |

1. Menghitung Ciri Turunan

Menghitung nilai ciri turunan fitur morfologi digital dihitung berdasarkan nilai ciri dasar pada tabel 4.15. Terdapat 6 ciri turunan yang digunakan pada penelitian ini, yaitu *aspect ratio*, *form factor*, *rectangularity*, *narrow factor*, *perimeter ration of diameter*, dan *perimeter ratio of physiological length* *and perimeter ratio of physiological width*.

Pada perhitungan nilai *aspect ratio* akan menggunakan Persamaan (2.51)

Pada perhitungan nilai *form factor* akan menggunakan Persamaan (2.52)

Pada perhitungan nilai *rectangularity* akan menggunakan Persamaan (2.53)

Pada perhitungan nilai *narrow factor* akan menggunakan Persamaan (2.54)

Pada perhitungan nilai *perimeter ration of diameter* akan menggunakan Persamaan (2.53)

Pada perhitungan nilai *perimeter ratio of physiological length* dan *perimeter ratio of physiological width* akan menggunakan Persamaan (2.56)

1. Nilai Ciri Fitur Morfologi

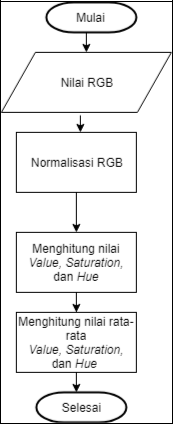
Berdasarkan perhitungan nilai turunan sebelumnya, maka didapatkan nilai fitur morfologi digital yaitu *Aspect Ratio* = 1,6115 ; *Form Factor* = 0,6410 ; *Recatangularity* = 1,3706 ; *Narrow Factor* = 0,7593 ; *Perimeter of Diameter* = 3,9239 ; *Perimeter ratio of L & W* = 1,8385.

1. Normalisasi Data

Normalisasi data dilakukan dengan menggunakan kedua nilai fitur ekstrasi yaitu GLCM dan morfologi digital yang telah dihitung sebelumnya. Normalisasi ini dihitung menggunakan Persamaan (2.57). Berikut Tabel 4.16 Nilai fitur sebelum normalisasi.

1. Ekstraksi fitur HSV

Proses ekstraksi ciri warna ini dimulai dari menormalisasi nilai RGB, mencari nilai HSV, dan terakhir mencari nilai *mean* dari nilai HSV tersebut. *Flowchart.*

**

**Gambar 4. 6 *Flowchart* Ekstraksi Warna HSV**

1. Nilai Red Green Blue (RGB)

Proses pertama pada metode HSV ini adalah mendapatkan nilai RGB pada citra dengan piksel 200x300, gambar 4.5 mempunyai nilai RGB pada Tabel 4.1, Tabel 4.2 dan Tabel 4.3.

1. Normalisasi nilai *Red Green Blue* (RGB)

Setelah mendapatkan nilai RGB, Langkah selanjutnya yang dilakukan pada metode HSV adalah menormalisasi nilai RGB dengan rumus 2.22. Berikut contoh perhitungan nilai normalisasi RGB, dengan mengambil beberapa contoh piksel yaitu piksel (1,1), (98,99), (99,99), (100,99), (300,200).

**a.** red (r)

Berdasarkan perhitungan diatas menormalisasi nilai R menjadi r, berikut tabel pada normalisasi red (r):

**Tabel 4. 4 Normalisasi r (red)**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x/y | 1 | ... | 98 | 99 | 100 | 101 | 102 | 103 | 104 | ... | 200 |
| 1 | 0.32026 |  | 0.32045 | 0.32045 | 0.32045 | 0.32045 | 0.32045 | 0.32045 | 0.32045 |  | 0.32385 |
| ... | ... | ... | ... | ... | … | … | ... | ... | ... | ... | ... |
| 98 | 0.31969 | ... | 0.24000 | 0.18750 | 0.19565 | 0.23770 | 0.23932 | 0.22222 | 0.20238 | ... | 0.32394 |
| 99 | 0.31969 | ... | 0.23333 | 0.21176 | 0.24324 | 0.26950 | 0.25688 | 0.25490 | 0.25263 | ... | 0.32394 |
| 100 | 0.31969 | ... | 0.27083 | 0.26471 | 0.27559 | 0.28873 | 0.29412 | 0.30252 | 0.30000 | ... | 0.32339 |
| 101 | 0.31969 | ... | 0.31250 | 0.30709 | 0.30935 | 0.31159 | 0.32353 | 0.32773 | 0.33663 | ... | 0.32334 |
| 102 | 0.31969 | ... | 0.33871 | 0.32836 | 0.33099 | 0.33333 | 0.35484 | 0.34899 | 0.35099 | ... | 0.32334 |
| 103 | 0.31969 | ... | 0.35433 | 0.34483 | 0.35043 | 0.35606 | 0.36290 | 0.35882 | 0.36129 | ... | 0.32334 |
| 104 | 0.31969 |  | 0.36800 | 0.35878 | 0.37190 | 0.36719 | 0.36601 | 0.36943 | 0.36571 |  | 0.32070 |
| ... | ... | ... | ... | ... | ... | … | ... | ... | ... | ... | ... |
| 300 | 0.32039 | ... | 0.32367 | 0.32367 | 0.32367 | 0.32367 | 0.32367 | 0.32367 | 0.32367 | ... | 0.32609 |

1. green (g)

Berdasarkan perhitungan diatas menormalisasi nilai G menjadi g, berikut tabel pada normalisasi *green* (g):

**Tabel 4. 5 Normalisasi *green* (g)**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x/y | 1 | ... | 98 | 99 | 100 | 101 | 102 | 103 | 104 | ... | 200 |
| 1 | 0.33660 |  | 0.33655 | 0.33655 | 0.33655 | 0.33655 | 0.33655 | 0.33655 | 0.33655 |  | 0.33649 |
| ... | ... | ... | ... | ... |  |  | ... | ... | ... | ... | ... |
| 98 | 0.33386 | ... | 0.52000 | 0.58750 | 0.55435 | 0.49180 | 0.48718 | 0.51515 | 0.55952 | ... | 0.33490 |
| 99 | 0.33386 | ... | 0.54444 | 0.57647 | 0.51351 | 0.46809 | 0.48624 | 0.49020 | 0.51579 | ... | 0.33490 |
| 100 | 0.33386 | ... | 0.53125 | 0.52941 | 0.49606 | 0.47183 | 0.47059 | 0.46218 | 0.50000 | ... | 0.33438 |
| 101 | 0.33386 | ... | 0.50000 | 0.48819 | 0.48201 | 0.47826 | 0.49020 | 0.46218 | 0.49505 | ... | 0.33438 |
| 102 | 0.33386 | ... | 0.47581 | 0.47761 | 0.47887 | 0.46939 | 0.51613 | 0.42953 | 0.43709 | ... | 0.33438 |
| 103 | 0.33386 | ... | 0.46457 | 0.50000 | 0.50427 | 0.47727 | 0.45968 | 0.41765 | 0.43226 | ... | 0.33438 |
| 104 | 0.33386 |  | 0.44000 | 0.44275 | 0.44628 | 0.44531 | 0.41830 | 0.41401 | 0.40571 |  | 0.33649 |
| .. | ... | ... | ... | ... | ... |  | ... | ... | ... | ... | ... |
| 300 | 0.33657 | ... | 0.33655 | 0.33655 | 0.33655 | 0.33655 | 0.33655 | 0.33655 | 0.33655 | ... | 0.33445 |

1. *blue* (b)

Berdasarkan perhitungan diatas menormalisasi nilai B menjadi b, berikut tabel pada normalisasi *blue* (b):

**Tabel 4. 6 Normalisasi *blue* (b)**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x/y | 1 | ... | 98 | 99 | 100 | 101 | 102 | 103 | 104 | ... | 200 |
| 1 | 0.34314 |  | 0.34300 | 0.34300 | 0.34300 | 0.34300 | 0.34300 | 0.34300 | 0.34300 |  | 0.33965 |
| ... | ... | ... | ... | ... |  |  | ... | ... | ... | ... | ... |
| 98 | 0.34646 | ... | 0.24000 | 0.22500 | 0.25000 | 0.27049 | 0.27350 | 0.26263 | 0.23810 | ... | 0.34116 |
| 99 | 0.34646 | ... | 0.22222 | 0.21176 | 0.24324 | 0.26241 | 0.25688 | 0.25490 | 0.23158 | ... | 0.34116 |
| 100 | 0.34646 | ... | 0.19792 | 0.20588 | 0.22835 | 0.23944 | 0.23529 | 0.23529 | 0.20000 | ... | 0.34223 |
| 101 | 0.34646 | ... | 0.18750 | 0.20472 | 0.20863 | 0.21014 | 0.18627 | 0.21008 | 0.16832 | ... | 0.34227 |
| 102 | 0.34646 | ... | 0.18548 | 0.19403 | 0.19014 | 0.19728 | 0.12903 | 0.22148 | 0.21192 | ... | 0.34227 |
| 103 | 0.34646 | ... | 0.18110 | 0.15517 | 0.14530 | 0.16667 | 0.17742 | 0.22353 | 0.20645 | ... | 0.34227 |
| 104 | 0.34646 |  | 0.19200 | 0.19847 | 0.18182 | 0.18750 | 0.21569 | 0.21656 | 0.22857 |  | 0.34281 |
| .. | ... | ... | ... | ... | ... |  | ... | ... | ... | ... | ... |
| 300 | 0.34304 | ... | 0.33977 | 0.33977 | 0.33977 | 0.33977 | 0.33977 | 0.33977 | 0.33977 | ... | 0.33946 |

1. Hitung Nilai *Hue Saturation Value* (HSV)

Proses ini merupakan proses menghitung nilai *Values*, *Saturation*, dan *Hue* berdasarkan nilai RGB yang terlah dinormalisasi pada Tabel 4.4,Tabel 4.5 dan Tabel 4.6. Dapat diketahui nilai piksel dan nilai RGB yang sudah dinormalisasinya adalah:

piksel (1,1), (98,99), (99,99), (100,99), (300,200).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| r1,1 | =0.32026 | g1,1 | =0.33660 | b1,1 | =0.34314 |
| r98,99 | =0.18750 | g98,99 | =0.58750 | b98,99 | =0.22500 |
| r99,99 | =0.21176 | g99,99 | =0.57647 | b99,99 | =0.21176 |
| r100,99 | =0.26471 | g100,99 | =0.52941 | b100,99 | =0.20588 |
| r300,200 | =0.32609 | g300,200 | =0.33445 | b300,200 | =0.33946 |

Tahap pertama mencari nilai *Value* dengan rumus 2.8 menggunakan rgb yang sudah dinormalisasi:

V1,1 = max(r1,1, g1,1, b1,1)

= max(0.32026, 0.33660, 0.34314)

= 0.34314

V98,99 = max(r98,99, g98,99, b98,99)

= max(0.18750, 0.58750, 0.22500)

= 0.58750

V 99,99 = max(r99,99, g99,99, b99,99)

= max(0.21176, 0.57647, 0.21176)

= 0.57647

V100,99 = max(r100,99, g100,99, b100,99)

= max(0.26471, 0.52941, 0.20588)

= 0.52941

V300,200 = max(r300,200, g300,200, b300,200)

= Max(0.32609, 0.33445, 0.33946)

= 0.33946

Nilai Value

**Tabel 4. 6 *Value* (V)**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x/y | 1 | ... | 98 | 99 | 100 | 101 | 102 | 103 | 104 | ... | 200 |
| 1 | 0.34314 |  | 0.34300 | 0.34300 | 0.34300 | 0.34300 | 0.34300 | 0.34300 | 0.34300 |  | 0.33965 |
| ... | ... | ... | ... | ... |  |  | ... | ... | ... | ... | ... |
| 98 | 0.34646 | ... | 0.52000 | 0.58750 | 0.55435 | 0.49180 | 0.48718 | 0.51515 | 0.55952 | ... | 0.34116 |
| 99 | 0.34646 | ... | 0.54444 | 0.57647 | 0.51351 | 0.46809 | 0.48624 | 0.49020 | 0.51579 | ... | 0.34116 |
| 100 | 0.34646 | ... | 0.53125 | 0.52941 | 0.49606 | 0.47183 | 0.47059 | 0.46218 | 0.50000 | ... | 0.34223 |
| 101 | 0.34646 | ... | 0.50000 | 0.48819 | 0.48201 | 0.47826 | 0.49020 | 0.46218 | 0.49505 | ... | 0.34227 |
| 102 | 0.34646 | ... | 0.47581 | 0.47761 | 0.47887 | 0.46939 | 0.51613 | 0.42953 | 0.43709 | ... | 0.34227 |
| 103 | 0.34646 | ... | 0.46457 | 0.50000 | 0.50427 | 0.47727 | 0.45968 | 0.41765 | 0.43226 | ... | 0.34227 |
| 104 | 0.34646 |  | 0.44000 | 0.44275 | 0.44628 | 0.44531 | 0.41830 | 0.41401 | 0.40571 |  | 0.34281 |
| .. | ... | ... | ... | ... | ... |  | ... | ... | ... | ... | ... |
| 300 | 0.34304 | ... | 0.33977 | 0.33977 | 0.33977 | 0.33977 | 0.33977 | 0.33977 | 0.33977 | ... | 0.33946 |

Setelah mendapatkan nilai value maka dicari nilai saturation dengan rumus 2.9. Perhitungan nilai S (Saturation)

**Tabel 4. 6 *Saturation* (S)**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x/y | 1 | ... | 98 | 99 | 100 | 101 | 102 | 103 | 104 | ... | 200 |
| 1 | 0.06667 |  | 0.06573 | 0.06573 | 0.06573 | 0.06573 | 0.06573 | 0.06573 | 0.06573 |  | 0.04651 |
| ... | ... | ... | ... | ... |  |  | ... | ... | ... | ... | ... |
| 98 | 0.07727 | ... | 0.53846 | 0.68085 | 0.64706 | 0.51667 | 0.50877 | 0.56863 | 0.63830 | ... | 0.05046 |
| 99 | 0.07727 | ... | 0.59184 | 0.63265 | 0.52632 | 0.43939 | 0.47170 | 0.48000 | 0.55102 | ... | 0.05046 |
| 100 | 0.07727 | ... | 0.62745 | 0.61111 | 0.53968 | 0.49254 | 0.50000 | 0.49091 | 0.60000 | ... | 0.05505 |
| 101 | 0.07727 | ... | 0.62500 | 0.58065 | 0.56716 | 0.56061 | 0.62000 | 0.54545 | 0.66000 | ... | 0.05530 |
| 102 | 0.07727 | ... | 0.61017 | 0.59375 | 0.60294 | 0.57971 | 0.75000 | 0.48438 | 0.51515 | ... | 0.05530 |
| 103 | 0.07727 | ... | 0.61017 | 0.68966 | 0.71186 | 0.65079 | 0.61404 | 0.46479 | 0.52239 | ... | 0.05530 |
| 104 | 0.07727 |  | 0.56364 | 0.55172 | 0.59259 | 0.57895 | 0.48438 | 0.47692 | 0.43662 |  | 0.06452 |
| .. | ... | ... | ... | ... | ... |  | ... | ... | ... | ... | ... |
| 300 | 0.06604 | ... | 0.04739 | 0.04739 | 0.04739 | 0.04739 | 0.04739 | 0.04739 | 0.04739 | ... | 0.03941 |

Setelah mendapatkan nilai *saturation* maka dicari nilai *hue* dengan rumus 2.10 karena perhitungan nilai value di atas mempunyai nilai yang sama dengan R yang telah dinormalisasi sehingga rumus dalam mencari H menggunakan rumus V=r. Perhitungan nilai *Hue* (H):

**Tabel 4. 6 *Hue* (H)**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x/y | 1 | ... | 98 | 99 | 100 | 101 | 102 | 103 | 104 | ... | 200 |
| 1 | 197.14286 |  | 197.14286 | 197.14286 | 197.14286 | 197.14286 | 197.14286 | 197.14286 | 197.14286 |  | 192.00000 |
| ... | ... | ... | ... | ... |  |  | ... | ... | ... | ... | ... |
| 98 | 208.23529 | ... | 120.00000 | 125.62500 | 129.09091 | 127.74194 | 128.27586 | 128.27586 | 126.00000 | ... | 201.81818 |
| 99 | 208.23529 | ... | 117.93103 | 120.00000 | 120.00000 | 117.93103 | 120.00000 | 120.00000 | 115.55556 | ... | 201.81818 |
| 100 | 208.23529 | ... | 106.87500 | 109.09091 | 109.41176 | 107.27273 | 105.00000 | 102.22222 | 100.00000 | ... | 205.00000 |
| 101 | 208.23529 | ... | 96.00000 | 98.33333 | 97.89474 | 97.29730 | 92.90323 | 92.00000 | 89.09091 | ... | 205.00000 |
| 102 | 208.23529 | ... | 88.33333 | 91.57895 | 90.73171 | 90.00000 | 85.00000 | 83.22581 | 82.94118 | ... | 205.00000 |
| 103 | 208.23529 | ... | 83.33333 | 87.00000 | 85.71429 | 83.41463 | 80.57143 | 78.18182 | 78.85714 | ... | 205.00000 |
| 104 | 208.23529 |  | 77.41935 | 80.62500 | 76.87500 | 78.18182 | 75.48387 | 73.54839 | 73.54839 |  | 197.14286 |
| .. | ... | ... | ... | ... | ... |  | ... | ... | ... | ... | ... |
| 300 | 197.14286 | ... | 192.00000 | 192.00000 | 192.00000 | 192.00000 | 192.00000 | 192.00000 | 192.00000 | ... | 202.50000 |

Proses selanjutnya adalah mencari nilai statistika ciri yaitu mencari nilai means dari nilai HSV yang telah didapat seperti perhitungan diatas dengan rumus 2.11. Berikut adalah perhitungan niali means dari nilai HSV yang telah diperoleh:

Tabel mean h s v seluruh data

|  |  |  |  |
| --- | --- | --- | --- |
| No | Mean H | Mean S | Mean V |
| 1 | 176.27642 | 0.13198 | 0.35885 |
| 2 | 182.89569 | 0.13553 | 0.36198 |
| 3 | 182.64133 | 0.11598 | 0.35605 |
| 4 | 182.51649 | 0.14819 | 0.36591 |
| 5 | 178.50649 | 0.14081 | 0.36123 |
| … | … | … | … |
| … | … | … | … |
| 96 | 172.99664 | 0.14310 | 0.36225 |
| 97 | 183.07471 | 0.15928 | 0.36971 |
| 98 | 182.89561 | 0.15879 | 0.36925 |
| 99 | 181.61706 | 0.12629 | 0.36236 |
| 100 | 176.27642 | 0.13198 | 0.35885 |

1. Normalisasi Data

Normalisasi data dilakukan dengan menggunakan kedua nilai fitur ekstrasi yaitu GLCM dan morfologi digital yang telah dihitung sebelumnya. Normalisasi ini dihitung menggunakan Persamaan (2.57). Berikut Tabel 4.16 Nilai fitur sebelum normalisasi.

Tabel 4.16 Nilai fitur data latih sebelum normalisasi

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Data Ke-** | **Form Factor** | **Aspect Ratio** | **Rect** | **Narrow Factor** | **Prd** | **Plw** | **Mean H** | **Mean S** | **Mean V** | **Kelas** |
| 1 | 2 | 0.5537 | 1.4871 | 1.3717 | 0.7901 | 4.2218 | 1.9944 | 182.8956885 | 0.135534714 | 0.361975814 | 0 |
| 2 | 3 | 0.6141 | 1.4374 | 1.3464 | 0.8111 | 4.009 | 1.9176 | 182.6413329 | 0.115976456 | 0.356051869 | 0 |
| 3 | 5 | 0.6249 | 1.5772 | 1.336 | 0.7773 | 3.9742 | 1.8906 | 178.5064888 | 0.14080869 | 0.361230077 | 0 |
| 4 | 6 | 0.5884 | 1.4592 | 1.3729 | 0.7972 | 4.0955 | 1.9374 | 181.6376174 | 0.127652966 | 0.357380103 | 0 |
| 5 | 7 | 0.5704 | 1.6124 | 1.369 | 0.7595 | 4.1595 | 1.9499 | 178.1473527 | 0.13541805 | 0.361303469 | 0 |
| … | … | … | … | … | … | … | … | … | … | … | … |
| … | … | … | … | … | … | … | … | … | … | … | … |
| 96 | 95 | 0.6008 | 1.3672 | 1.3678 | 0.8251 | 4.0532 | 1.9316 | 174.0208848 | 0.143093 | 0.363129952 | 1 |
| 97 | 96 | 0.4791 | 1.4476 | 1.4292 | 0.7845 | 4.5389 | 2.1059 | 173.0916171 | 0.14369116 | 0.362330018 | 1 |
| 98 | 97 | 0.4752 | 1.4469 | 1.4289 | 0.7847 | 4.5574 | 2.1148 | 172.996641 | 0.14309828 | 0.362251019 | 1 |
| 99 | 98 | 0.6189 | 1.3509 | 1.3552 | 0.8339 | 3.9934 | 1.9137 | 183.0747065 | 0.159276869 | 0.369708539 | 1 |
| 90 | 99 | 0.6139 | 1.349 | 1.3545 | 0.8348 | 4.0094 | 1.9221 | 182.8956138 | 0.158786648 | 0.36925264 | 1 |

Berdasarkan tabel diatas, maka perlu untuk menentukan nilai minimum dan maksimum setiap nilai fitur yang berguna pada perhitungan normalisasi nantinya. Berikut Tabel 4.17 Nilai minimum dan maksimum dari setiap nilai fitur sebelum normalisasi.

Tabel 4.17 Nilai Minimum dan Maksimum Fitur Morfologi dan HSV Data Latih

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **NILAI MINIMUM DAN MAKSIMUM** | | | | | | | | | | | | | | | | | |
| **No** | Form Factor | Aspect Ratio | Rect | | Narrow  Factor | | Prd | Plw | Hue | | Saturation | | Value | | | Kelas | |
| **Nilai Minimum Fitur Morfologi** | | | | | | | | | **Nilai Minimum Fitur HSV** | | | | | | | |  |
| **1** | 0.18 | 1.07180 | 1.28240 | | 0.25340 | | 3.58590 | 1.52920 | 167.13437 | | 0.09674 | | 0.35257 | | | 0 | |
| **Nilai Maksimum Fitur Morfologi** | | | | | | | | | **Nilai Maksimum Fitur HSV** | | | | | | | |  |
| **2** | 0.76750 | 9.48580 | 2.61140 | 0.96200 | | 7.40480 | | 2.49990 | | 191.14900 | | 0.21489 | | 0.38165 | 1 | | |

Berdasarkan Tabel 4.17 diatas, nilai minimum dan maksimum yang telah diketahui akan digunakan pada perhitungan normalisasi data fitur GLCM dan morfologi digital menggunakan Persamaan (2.58). Berikut contoh perhitungan pada data ke-100.

Tabel 4.18 Nilai Fitur Data Latih Setelah Normalisasi

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Data Ke-** | **Form Factor** | **Aspect Ratio** | **Rect** | **Narrow Factor** | **Prd** | **Plw** | **Mean H** | **Mean S** | **Mean V** | **Kelas** |
| 1 | 2 | 0.63609 | 0.04936 | 0.06719 | 0.75741 | 0.16651 | 0.47924 | 0.65632 | 0.32837 | 0.32343 | 0 |
| 2 | 3 | 0.73889 | 0.04345 | 0.04816 | 0.78704 | 0.11079 | 0.40012 | 0.64573 | 0.16284 | 0.11976 | 0 |
| 3 | 5 | 0.75728 | 0.06007 | 0.04033 | 0.73935 | 0.10168 | 0.37231 | 0.47355 | 0.37300 | 0.29779 | 0 |
| 4 | 6 | 0.69515 | 0.04604 | 0.06810 | 0.76743 | 0.13344 | 0.42052 | 0.60393 | 0.26166 | 0.16542 | 0 |
| 5 | 7 | 0.66451 | 0.06425 | 0.06516 | 0.71423 | 0.15020 | 0.43340 | 0.45859 | 0.32738 | 0.30031 | 0 |
| … | … | … | … | … | … | … | … | … | … | … | … |
| … | … | … | … | … | … | … | … | … | … | … | … |
| 96 | 95 | 0.71626 | 0.03511 | 0.06426 | 0.80680 | 0.12237 | 0.41455 | 0.28676 | 0.39234 | 0.36311 | 1 |
| 97 | 96 | 0.50911 | 0.04466 | 0.11046 | 0.74951 | 0.24955 | 0.59411 | 0.24807 | 0.39740 | 0.33560 | 1 |
| 98 | 97 | 0.50247 | 0.04458 | 0.11023 | 0.74979 | 0.25439 | 0.60328 | 0.24411 | 0.39238 | 0.33289 | 1 |
| 99 | 98 | 0.74706 | 0.03317 | 0.05478 | 0.81922 | 0.10671 | 0.39611 | 0.66378 | 0.52931 | 0.58929 | 1 |
| 90 | 99 | 0.73855 | 0.03295 | 0.05425 | 0.82049 | 0.11090 | 0.40476 | 0.65632 | 0.52516 | 0.57361 | 1 |

Tabel 4.18 Nilai Fitur Data Uji Setelah Normalisasi

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Data Ke-** | **Form Factor** | **Aspect Ratio** | **Rect** | **Narrow Factor** | **Prd** | **Plw** | **Mean H** | **Mean S** | **Mean V** | **Kelas** |
| 1 | 1 | 0.61974 | 0.05358 | 0.08254 | 0.73624 | 0.17628 | 0.47718 | 0.38069 | 0.29829 | 0.21580 | 0 |
| 2 | 4 | 0.65532 | 0.05464 | 0.05320 | 0.74866 | 0.15541 | 0.46636 | 0.64053 | 0.43546 | 0.45879 | 0 |
| 3 | 45 | 0.66979 | 0.06451 | 0.06531 | 0.71338 | 0.14729 | 0.42753 | 0.42434 | 0.35930 | 0.32977 | 0 |
| 4 | 47 | 0.64102 | 0.04981 | 0.06742 | 0.75572 | 0.16366 | 0.47306 | 0.61650 | 0.35590 | 0.35582 | 0 |
| 5 | 48 | 0.63677 | 0.04938 | 0.06652 | 0.75769 | 0.16617 | 0.47924 | 0.61935 | 0.35390 | 0.34999 | 0 |
| 6 | 51 | 0.57872 | 0.04388 | 0.08337 | 0.76630 | 0.20179 | 0.53456 | 0.78037 | 0.27447 | 0.28370 | 1 |
| 7 | 56 | 0.75557 | 0.04602 | 0.02483 | 0.79184 | 0.10252 | 0.40486 | 0.94678 | 0.27647 | 0.40250 | 1 |
| 8 | 87 | 0.69549 | 0.03848 | 0.04033 | 0.80864 | 0.13331 | 0.45555 | 0.73262 | 0.37580 | 0.47714 | 1 |
| 9 | 89 | 0.72613 | 0.03388 | 0.06125 | 0.81301 | 0.11721 | 0.40898 | 0.35397 | 0.33662 | 0.31162 | 1 |
| 10 | 100 | 0.41974 | 0.05813 | 0.11467 | 0.70646 | 0.32059 | 0.70197 | 0.60308 | 0.25012 | 0.33675 | 1 |

### Klasifikasi

Klasifikasi pada penelitian menggunakan metode ANFIS. Input yang digunakan pada tahap klasifikasi ini adalah nilai fitur yang telah dinormalisasi sebelumnya pada tahap *processing*. Pada metode ini nilai fitur yang telah dinormalisasi akan dibagi menjadi 2 data, yaitu data latih dan data uji. Data-data tersebut akan digunakan pada proses pelatihan dan pengujian. Berikut Gambar 4.13 *flowchart* proses pelatihan pada ANFIS:



Gambar 4.13 Proses Pelatihan pada ANFIS

Pada proses pelatihan ini terdapat tahap *forward* (maju) dan *backward* (mundur). Data latih yang telah dibagi sebanyak 80 citra. Kemudian, dilakukan proses pembentukan fungsi keanggotaan *fuzzy* yang akan mendapatkan nilai paramter , , dan . Setelah fungsi keanggotaan *fuzzy* terbentuk, data latih akan dihitung pada tahap *forward* dari *layer* 1 sampai ke *layer* 5. Setelah perhitungan berada pada *layer* 5, sistem akan melakukan pemeriksaan jumlah *epoch* dan jumlah minimal *error*. Jika kedua kondisi tersebut belum terpenuhi, maka sistem akan melakukan tahap *backward* yaitu perhitungan *error* dengan menggunakan algoritma EBP (*Error Backprogation*). Sistem akan memperbarui parameter dan dengan menggunakan model propagasi *error gradient descent*. Kemudian, jika kedua kondisi tersebut telah terpenuh, sistem akan mendapatkan *fuzzy inference system* yang terdiri dari nilai keanggotaan *fuzzy* dan parameter konstan aturan *fuzzy*. Berikut Tabel 4.19 Data latih sebanyak 80 nilai fitur.

Tabel 4.19 Data Latih

Data latih ini adalah hasil pembagian data pada nilai fitur yang telah dinormalisasi, data lagi yang digunakan sebanyak 80 nilai fitur. Kemudian ke 10 data lainnya akan digunakan pada prses pengujian. Berikut *flowchart* Gambar 4.14 proses pengujian pada ANFIS:



Gambar 4.14 Proses Pengujian pada ANFIS

Pada proses ini digunakan 20 data sebagai masukan untuk diuji. *Fuzzy inference system* yang telah didapatkan pada proses pelatihan akan digunakan pada proses pengujian untuk mendapatkan hasil klasifikasi. Berikut Tabel 4.20 Data uji sebanyak 20 nilai fitur.

Tabel 4.20 Data Uji

#### Pembentukan Fungsi Keanggotaan Fuzzy

Pembentukan fungsi keanggotaan nilai *fuzzy* ini dilakukan pada proses pelatihan dengan menjadikan 11 fitur sebagai *input* atau masukan. Pada tahap pembentukan FIS ini akan ditentukan jumlah himpunan menggunakan *subtractive clustering*.

1. ***Subtractive Clustering***

Tahapan ini bertujuan untuk menentukan *cluster* pusat atau *center cluster.* Data yang akan digunakan untuk menentukan *center cluster* ini adalah data latih dan termasuk nilai kelasnya yang telah dinormalisasi. Kelas yang akan dinormalisasikan dihitung menggunakan Persamaan (2.56) sebagai berikut:

Berikut Tabel 4.21 Data Latih ke-1 yang telah dinormaslisasikan.

Tabel 4.21 Data Latih ke-1 beserta kelas

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Nilai Fitur Morfologi** | | | | | | **Nilai Fitur HSV** | | |  |
| Form Factor | Aspect Ratio | Rect | Narrow  Factor | Prd | Plw | H | S | V | Kelas |
| 1 | 0.63609 | 0.04936 | 0.06719 | 0.75741 | 0.16651 | 0.47924 | 0.65632 | 0.32837 | 0.32343 | 0 |

Setelah itu, seluruh data beserta kelasnya yang telah dinormalisasikan akan digunakan untuk menentukan *potential value* dan *center cluster*, ada berapa tahapan untuk mencari kedua nilai tersebut yakni sebagai berikut:

* + - 1. *Potential Value*

*Potential value* ini bertujuan untuk mencari data yang memiliki potensial tertinggi untuk dijadikan *center* *cluster*. Berikut ini contoh tahapan perhitungan *potential value* data ke-1.

1. Menentukan Nilai dan

Perhitungan potensial value data latih ke-i dilakukan dengan melibatkan seluruh data latih.. Nilai dari data ke-i yang disimbolkan dikurangi dengan setiap nilai data ,. contohnya seperti Perhitungan dengan persamaan berikut.

; [0.32343-0.32343] ; [0-0]

= [0] ; [0 ];;[ 0 ];[ 0]

; [0.32343- 0.57361] ; [0-1]

= [-0.10246]; [0.01641];;[ -0.25018] ; [-1]

= [-0.20492 – (-0.20492)];[0.03282 - 0.03282];…;[ -0.50036 – (-0.50036)];[-2 – (-2)]

1. Menghitung *Potential Value*

Berdasarkan nilai , nilai-nilai tersebut akan digunakan untuk menghitung nilai *potential value* menggunakan Persamaan (2.4) berikut ini.

Berdasarkan perhitungan diatas telah didapat nilai *potential value*, terlihat Persamaan (2.4) dihitung secara berulang dengan menggunakan satu baris data latih sebagai nilai hingga seluruh data digunakan untuk mendapatkan 90 *potential value*. Berikut tabel 4.25 nilai *potential value* yang telah dihitung.

Tabel 4.25 Nilai *Potential Value*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Nilai Potential Value | | | | | |
|  |  |  |  |  |  |
| 1 | 9.61870 | 6.04091 |  | 4.59010 | 4.57848 | 4.79474 |

Nilai *potential value* yang telah diketahui akan ditentukan nilai maksimum *potential value* dan nilai referensi maksimum *potential value* berdasarkan nilai tertinggi sebagai berikut:

Dan

Nilai *maxPotval* ini diketahui berada diurutan data ke-15 sehingga dapat disimpulkan nilai *center cluster* yang pertama yakni berada dititik data ke-15. Berikut tabel 4.26 nilai center cluster ke-1.

Tabel 4.26 Nilai *Center Cluster* ke-1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Data ke** | **Nilai Fitur Morfologi** | | | | | | | **Nilai Fitur GLCM** | | | | | **kelas** |
| **177** | 0.70451 | 0.07611 | 0.02566 | 0.70308 | 0.12844 | 0.41712 | 0.06937 | | 0.88315 | 0.90351 | 0.70451 | 0.07611 | 0 |

* + - 1. *Center cluster*

Nilai *center cluster* ini adalah nilai yang telah dikelompokan berdasarkan tingginya nilai potensial yang telah didapat. Berikut tahapan perhitungan *center cluster* ke-*n*.

1. Menentukan nilai dan

Nilai yang digunakan pada Persamaan (2.5) ini adalah nilai *center cluster* pertamayang telah didapat seperti yang terlihat pada tabel 4.26, Sedangkan nilai nya masih menggunakan data pada tabel 4.19 diatas.

Kemudian nilai dikurangi dengan setiap nilai , berikut hasil pengurangannya dapat dilihat pada tabel 4.27 berikut ini.

Tabel 4.27 Nilai dan baru

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Nilai**  dan baru | | | | | | | | | **kelas** |
| **1** | 0.06842 | 0.02675 | -0.04153 | -0.05433 | -0.03807 | -0.06212 | -0.58695 | 0.55478 | 0.58008 | 0 |
| **2** | -0.03438 | 0.03266 | -0.02250 | -0.08396 | 0.01765 | 0.01700 | -0.57636 | 0.72031 | 0.78375 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **47** | 0.46604 | 0.03174 | -0.15184 | -0.01439 | -0.38653 | -0.58288 | -0.71608 | 0.38808 | 0.40538 | -1 |
| **48** | 0.66434 | -0.92389 | -0.30737 | 0.66695 | -0.75574 | 0.18244 | -0.77704 | 0.58796 | 0.49899 | -1 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **89** | -0.04255 | 0.04294 | -0.02912 | -0.11614 | 0.02173 | 0.02101 | -0.59441 | 0.35384 | 0.31422 | -1 |
| **90** | -0.03404 | 0.04316 | -0.02859 | -0.11741 | 0.01754 | 0.01236 | -0.58695 | 0.35799 | 0.32990 | -1 |

1. Menghitung nilai revisi *potential value*

Perhitungan pada persamaan (2.5), untuk mencari nilai *center cluster* ke-*n* ini. Data potensial yang akan digunakan sesuai pada tabel 4.25 yang telah dihitung dengan nilai maksimum potensialnya yaitu . Apabila nilai lebih kecil dari 0 maka nilai diubah menjadi 0. Berikut perhitungan nilai revisi potensial dibawah ini.

Berdasarkan perhitungan diatas maka didapatlah nilai potensial baru. Berikut tabel 4.28 Nilai revisi *potential value*

Tabel 4.28 Nilai revisi *potential value*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Nilai Revisi Potential Value | | | | | |
|  |  |  |  |  |  |
| 1 | 9.61833 | 6.04091 |  | 4.59010 | 4.57848 | 4.79474 |

Nilai revisi *potential value* ini yang telah didapat sesuai dengan tabel 4.28 maka akan ditentukan kembali nilai maksimum *potential value* sebagai berikut:

Nilai maksimum *potential value* ini diketahui berada diurutan data ke-26 sehingga dapat disimpulkan nilai *center cluster* kedua yakni berada dititik data ke-26.

Pencarian *center cluster* selanjutnya dilakukan apabila nilai *ratio* = *accept ratio* dan *reject ratio*, jika nilai *ratio* < *accept ratio* dan *reject ratio* maka pencarian *center cluster* akan dihentikan. Berikut tabel 4.29 nilai *center cluster* yang dihitung berdasarkan perhitungan diatas.

Tabel 4.29 Nilai *center cluster*

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Data Berpotensi** | **Nilai *center cluster*** | | | | | | | | | kelas |
| 1 | 15 | 0.70451 | 0.07611 | 0.02566 | 0.70308 | 0.12844 | 0.41712 | 0.06937 | 0.88315 | 0.90351 | 0 |
| 2 | 26 | 0.67353 | 0.06427 | 0.06501 | 0.71423 | 0.14523 | 0.42423 | 0.45834 | 0.32761 | 0.30077 | 0 |
| 3 | 82 | 0.75983 | 0.07064 | 0.03258 | 0.7138 | 0.1004 | 0.36479 | 0.7567 | 0.18649 | 0.22477 | 1 |
| 4 | 80 | 0.51013 | 0.04487 | 0.11061 | 0.7488 | 0.24871 | 0.59225 | 0.28293 | 0.35933 | 0.29819 | 1 |
| 5 | 78 | 0.74298 | 0.03274 | 0.05418 | 0.8212 | 0.1087 | 0.40084 | 0.70935 | 0.48432 | 0.53084 | 1 |
| 6 | 34 | 0.62894 | 0.06348 | 0.05553 | 0.72142 | 0.17078 | 0.48182 | 0.14167 | 0.71126 | 0.64265 | 0 |
| 7 | 77 | 0.44409 | 0.08594 | 0.23085 | 0.58524 | 0.29996 | 0.51592 | 0.88719 | 0.22758 | 0.35072 | 1 |
| 8 | 47 | 0.23847 | 0.04437 | 0.1775 | 0.71747 | 0.51497 | 1 | 0.78545 | 0.49507 | 0.49813 | 1 |
| 9 | 48 | 0.04017 | 1 | 0.33303 | 0.03613 | 0.88418 | 0.23468 | 0.84641 | 0.29519 | 0.40452 | 1 |

#### Pelatihan ANFIS

Pelatihan pada ANFIS ini menggunakan data latih pada berdasarkan Tabel 4.19 sebagai *input*. Terdapat 5 *layer* yang akan dijalankan pada proses pelatihan ini. Pada *layer* 1 terdapat 9 variabel *input* yang memiliki 9 himpunan *fuzzy* yang membentuk 81 neuron. Pada *layer* 2, 3, dan 4 akan digunakan sebanyak 9 rule sesuai dengan jumlah *center cluster*. Sedangkan pada *layer* 5 terdapat 1 neuron *output*.

Perhitungan pada ANFIS ini akan menggunakan fungsi keanggotaan *gaussian*. Parameter yang dibutuhkan pada *gaussmf* ini adalah parameter dan *c* sesuai pada perhitungan pada Persamaan (2.3). Nilai *center cluster* pada tabel 4.29 akan digunakan sebagai parameter *c* sedangkan parameter akan dihitung menggunakan Persamaan (2.6) sebagai berikut:

1. **Tahap Maju (*forward*) ANFIS**

Pada tahap ini melakukan proses perhitungan dari *layer* 1 hingga *layer* 5. Sehingga pada tahap ini disebut tahap maju (*forward*). Berikut perhitungan setiap *layer*.

1. *Layer* 1

Pada *layer* 1 ini adalah proses memetakan masukan ke dalam himpunan *fuzzy* yang disebut dengan proses *fuzzyfication*. Perhitungan pada *layer* 1 ini akan menggunakan Persamaan (2.3) kurva *gaussian* pada fungsi keanggotaan *fuzzy*. *Layer* 1 ini menghitung setiap data yang terdiri 9 *input* dan setiap *input* terdiri dari 9 *mf*.

Data ke-1 input ke-1

Perhitungan diatas adalah nilai data ke-1, perhitungan tersebut dihitung hingga data ke-90, maka dapat dilihat Tabel 4.30 hasil perhitungan pada *layer* 1 berikut ini.

Tabel 4.30 Hasil Perhitungan *Layer* 1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Data ke-** | **Input** | **Mf** | | | | | | | |
| **1** | **2** | **3** | **4** | **...** | **7** | **8** | **9** |
| 1 | 1 | 0.92784 | 0.97782 | 0.78272 | 0.77580 | ... | 0.55442 | 0.07969 | 0.00341 |
| 2 | 0.98862 | 0.99645 | 0.99278 | 0.99968 | ... | 0.97882 | 0.99960 | 0.00000 |
| 3 | 0.97278 | 0.99992 | 0.98102 | 0.97029 | ... | 0.65145 | 0.82309 | 0.32280 |
| 4 | 0.95387 | 0.97061 | 0.97003 | 0.99881 | ... | 0.62233 | 0.97480 | 0.00024 |
| 5 | 0.97708 | 0.99278 | 0.93246 | 0.89753 | ... | 0.75206 | 0.14330 | 0.00026 |
| 6 | 0.94013 | 0.95274 | 0.81092 | 0.81519 | ... | 0.97870 | 0.01305 | 0.38406 |
| 7 | 0.00404 | 0.53412 | 0.85111 | 0.10745 | ... | 0.42621 | 0.76583 | 0.56094 |
| 8 | 0.00727 | 0.99999 | 0.72464 | 0.98478 | ... | 0.84998 | 0.64107 | 0.98254 |
| 9 | 0.00459 | 0.99182 | 0.85578 | 0.98986 | ... | 0.98815 | 0.61366 | 0.90014 |
| 2 | 1 | 0.98127 | 0.93393 | 0.99301 | 0.43288 | ... | 0.24895 | 0.01819 | 0.00041 |
| 2 | 0.98308 | 0.99309 | 0.98824 | 0.99997 | ... | 0.97153 | 0.99999 | 0.00000 |
| 3 | 0.99193 | 0.99547 | 0.99612 | 0.93951 | ... | 0.58625 | 0.76517 | 0.27296 |
| 4 | 0.89334 | 0.91868 | 0.91775 | 0.97687 | ... | 0.52123 | 0.92548 | 0.00012 |
| 5 | 0.99503 | 0.98120 | 0.99827 | 0.73760 | ... | 0.56408 | 0.07326 | 0.00007 |
| 6 | 0.99539 | 0.99074 | 0.98023 | 0.55398 | ... | 0.80690 | 0.00316 | 0.64537 |
| 7 | 0.00492 | 0.57016 | 0.82117 | 0.12173 | ... | 0.39343 | 0.73173 | 0.52500 |
| 8 | 0.00025 | 0.64766 | 0.99109 | 0.53917 | ... | 0.93514 | 0.17101 | 0.75558 |
| 9 | 0.00005 | 0.59201 | 0.83825 | 0.60086 | ... | 0.42593 | 0.10120 | 0.27324 |
|  |  |  |  |  |  | ... |  |  |  |
| 90 | 1 | 0.98163 | 0.93460 | 0.99278 | 0.43396 | ... | 0.24975 | 0.01829 | 0.00041 |
| 2 | 0.97064 | 0.98443 | 0.97753 | 0.99773 | ... | 0.95607 | 0.99792 | 0.00000 |
| 3 | 0.98701 | 0.99815 | 0.99251 | 0.95045 | ... | 0.60714 | 0.78423 | 0.28838 |
| 4 | 0.80207 | 0.83472 | 0.83350 | 0.92106 | ... | 0.41252 | 0.84383 | 0.00005 |
| 5 | 0.99509 | 0.98132 | 0.99824 | 0.73796 | ... | 0.56445 | 0.07336 | 0.00007 |
| 6 | 0.99756 | 0.99395 | 0.97476 | 0.56982 | ... | 0.82061 | 0.00345 | 0.62950 |
| 7 | 0.00404 | 0.53412 | 0.85111 | 0.10745 | ... | 0.42621 | 0.76583 | 0.56094 |
| 8 | 0.12867 | 0.53557 | 0.15959 | 0.64404 | ... | 0.24247 | 0.98562 | 0.42905 |
| 9 | 0.17528 | 0.30390 | 0.14270 | 0.29710 | ... | 0.45164 | 0.91288 | 0.63289 |

1. *Layer 2*

Pada *layer* 2 ini dilakukan perhitungan dengan melakukan perkalian derajat keanggotaan yang telah didapat pada *layer* 1. *Output* yang dihasilkan pada *layer* 2 ini disebut kekuatan aktivasi (*firing strength*). Contoh perhitungan pada *layer* 2 ini adalah data ke-1 di *layer* 1.

Perhitungan ini dilakukan terhadap keseluruh 90 data pada *layer* 1. Pada perhitungan diatas maka dapat dilihat Tabel 4.31 hasil perhitungan pada *layer* 2 berikut ini.

Tabel 4.31 Hasil Perhitungan *Layer* 2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Data ke-** | ***W*** | | | | | | | |
| **1** | **2** | **3** | **4** | **...** | **7** | **8** | **9** |
| **1** | 1.05279E-07 | 0.47382 | 0.29512 | 0.05760 | **...** | 0.05797 | 3.60061E-05 | 7.04106E-18 |
| **2** | 5.56857E-11 | 0.18025 | 0.59890 | 0.00640 | **...** | 0.00527 | 3.77487E-08 | 2.85822E-20 |
|  |  |  |  |  |  |  |  |  |
| **90** | 6.81793E-05 | 0.065 | 0.01514 | 0.00328 | **...** | 0.00129 | 2.10753E-06 | 1.33101E-20 |

1. *Layer* 3

Pada *layer* 3 ini setiap neuronnya bersifat non­-adaptif atau tetap. Contoh perhitungan pada data ke-1 di *layer* 3 ini menggunakan Persamaan (2.9) sebagai berikut.

Pada perhitungan diatas maka dapat dilihat Tabel 4.32 hasil perhitungan pada *layer* 3 berikut ini.

Tabel 4.32 Hasil Perhitungan *Layer* 3

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Data ke-** |  | | | | | | | |
| **1** | **2** | **3** | **4** | **...** | **7** | **8** | **9** |
| **1** | 9.55776E-08 | 0.43016 | 0.26792 | 0.05229 | **...** | 0.05262 | 3.2688E-05 | 6.39221E-18 |
| **2** | 6.93834E-11 | 0.22459 | 0.74622 | 0.00798 | **...** | 0.00657 | 4.70343E-08 | 3.56129E-20 |
|  |  |  |  |  |  |  |  |  |
| **90** | 6.86765E-05 | 0.06547 | 0.01525 | 0.0033 | **...** | 0.0013 | 2.1229E-06 | 1.34072E-20 |

1. *Layer* 4

Pada *layer* 4 ini adalah proses mengubah hasil *fuzzy* ke bentuk keluaran yang *crisp*, proses itu disebut dengan proses *defuzzyfication*. Perhitungan pada *layer* 4 ini menggunakan Persamaan (2.10). Variabel adalah hasil atau *output* dari *layer* 3 dan parameter konsekuen(). Parameter konsekuentersebut didapatkan menggunakan matriks yang dihitung sebanyak 90 data latih. Berikut perhitungan mendapatkan nilai parameter konsekuen.

Parameter dari matriks dengan target *output y* menggunakan perhitungan berikut.

Berikut nilai parameter konsekuen () dapat dilihat tabel 4.33 berikut ini.

Tabel 4.33 Parameter Konsekuen ()

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Rule ke- |  | | | | | | |
| *O* | *p* | *q* |  | *v* | *w* | *x* |
| 1 | -38.11742 | 861.47152 | 247.58306 |  | 2.88451 | -5.30225 | -255.14427 |
| 2 | -1767.88586 | -12271.02271 | -7740.65926 |  | 48.35863 | -215.37772 | 5399.88343 |
|  |  |  |  |  |  |  |  |
| 8 | 2447258.48538 | -20881371.13791 | -868351.41332 |  | -1523495.83033 | 1987416.96373 | 6384625.07346 |
| 9 | 3.39443 | -447758296129836 | -447758296129836 1.51601 |  | -2.11945 | -193139230501844 | 607348834733069 |

Setelah mengetahui nilai parameter konsekuen diatas. Berikut contoh perhitungan nilai *f* data ke-1 menggunakan Persamaan (2.13).

Perhitungan diatas akan mendapatkan nilai hingga pada data ke-1, dapat dilihat pada tabel 4.34 berikut ini.

Tabel 4.34 Nilai *f*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Data ke- |  | | | | | |
| 1 | 2 | 3 | ... | 8 | 9 |
| 1 | 5.936119166 | -81.83675063 | 30.03973131 | ... | -55571.93774 | -5.94654E+14 |
| 2 | 7.653395348 | -31.5622738 | 4.260116883 | ... | -201523.6293 | -3.06476E+14 |
|  |  |  |  | ... |  |  |
| 89 | 8.689746606 | -122.7944441 | 24.05102824 | ... | 64429.90723 | -1.32257E+15 |
| 90 | 8.918564447 | -119.0618221 | 23.04082247 | ... | 14703.51358 | -1.3052E+15 |

Hasil perhitungan nilai *f* diatas akan dikalikan dengan nilai pada layer 3. Berikut contoh perhitungan data ke-1 pada layer 4.

Pada perhitungan diatas maka dapat dilihat Tabel 4.35 hasil seluruh perhitungan pada *layer* 4 berikut ini.

Tabel 4.35 Hasil Perhitungan *Layer* 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data ke- |  | | | | |
| 1 | 2 | ... | 8 | 9 |
| 1 | 5.6736E-07 | -35.20291685 | ... | -1.8165359 | -0.003801148 |
| 2 | 5.31019E-10 | -7.088672199 | ... | -0.009478513 | -1.09145E-05 |
|  |  |  | ... |  |  |
| 89 | 0.000653857 | -6.552514344 | ... | 0.09823767 | -1.38852E-05 |
| 90 | 0.000612496 | -7.795285212 | ... | 0.031214136 | -1.7499E-05 |

1. *Layer* 5

Pada *layer* 5 ini adalah proses penjumlahan yang telah dihasilkan pada *layer* 4. Hasil perhitungan pada *layer* 5 ini juga menjadi keluaran / *output* tahap *forward* pada metode ANFIS. Perhitungan pada *layer* 5 ini menggunakan Persamaan (2.11).

Pada perhitungan diatas maka dapat dilihat Tabel 4.36 hasil perhitungan pada *layer* 5 berikut ini.

Tabel 4.36 Hasil Perhitungan *Layer* 5

|  |  |
| --- | --- |
| **Data** | **Hasil Perhitungan** |
| 1 | -4.8585E-11 |
| 2 | 0.085966062 |
| 3 | -6.55054E-08 |
| 4 | 4.37607E-07 |
|  |  |
| 88 | 1.000000027 |
| 89 | 0.999999927 |
| 90 | 1.000000052 |

1. **Tahap Mundur (*backward*) ANFIS**

Tahap *backward* ini dilakukan untuk menghitung *error* pada tiap *layer*. Tahap *backward* ini juga akan memperbarui nilai parameter dan pada fungsi keanggotaan menggunakan model propagasi *error gradient descent*. Perhitungan *error* pada tiap *layer* menggunakan algoritma EBP (*error backpropagation*).

1. *Error* *layer* 5

*Error* pada *layer* 5 ini hanya memiliki 1 neuron *output*. Perhitungan *error* pada *layer* 5 ini menggunakan Persamaan (2.13).

Pada perhitungan diatas maka dapat dilihat Tabel 4.37 hasil perhitungan *error* pada *layer* 5 berikut ini.

Tabel 4.37 *Error* *Layer* 5

|  |  |
| --- | --- |
| **Data** | **Hasil Perhitungan** |
| 1 | -9.71710E-11 |
| 2 | 1.71932E-01 |
| 3 | -1.31011E-07 |
| 4 | 8.75214E-07 |
|  |  |
| 88 | 1 |
| 89 | 1 |
| 90 | 1 |

1. *Error layer* 4

Propagasi *error* pada *layer* 4 ini bernilai sama dengan *error layer* 5 karena tahap *backward* bersifat tetap. Nilai *error* pada *layer* 4 ini dirumuskan pada Persamaan (2.14).

Pada perhitungan diatas maka dapat dilihat Tabel 4.38 hasil perhitungan *error* pada *layer* 5 berikut ini.

Tabel 4.38 Hasil Perhitungan *Error Layer* 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data ke- |  | | | | |
| 1 | 2 | ... | 8 | 9 |
| 1 | -9.7171E-11 | -9.7171E-11 | ... | -9.7171E-11 | -9.7171E-11 |
| 2 | 1.7193E-01 | 1.7193E-01 | ... | 1.7193E-01 | 1.7193E-01 |
|  |  |  | ... |  |  |
| 89 | -1.4594E-07 | -1.4594E-07 | ... | -1.4594E-07 | -1.4594E-07 |
| 90 | 1.0414E-07 | 1.0414E-07 | ... | 1.0414E-07 | 1.0414E-07 |

1. *Error layer* 3

Nilai *error* pada *layer* 3 ini dirumuskan pada Persamaan (2.15). nilai pada Persamaan (2.56) tersebut sama dengan nilai pada tabel yang digunakan di *layer* 4 pada tahap *forward*.

Pada perhitungan diatas maka dapat dilihat Tabel 4.39 hasil perhitungan *error* pada *layer* 4 berikut ini.

Tabel 4.39 Hasil Perhitungan *Error Layer* 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data ke- |  | | | | |
| 1 | 2 | ... | 8 | 9 |
| 1 | -5.7682E-10 | 7.9522E-09 | ... | 5.4000E-06 | 5.7783E+04 |
| 2 | 1.3159E+00 | -5.4266E+00 | ... | -3.4648E+04 | -5.2693E+13 |
|  |  |  | ... |  |  |
| 89 | -1.2682E-06 | 1.7921E-05 | ... | -9.4032E-03 | 1.9302E+08 |
| 90 | 9.2881E-07 | -1.2399E-05 | ... | 1.5313E-03 | -1.3593E+08 |

1. *Error layer* 2

Perhitungan yang akan digunakan pada *error layer* 2 ini berdasarkan Persamaan (2.18). Berikut perhitungan *error layer* 2.

Pada perhitungan diatas maka dapat dilihat Tabel 4.40 hasil perhitungan *error* pada *layer* 2 berikut ini.

Tabel 4.40 Error Layer 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data ke- |  | | | | |
| 1 | 2 | ... | 8 | 9 |
| 1 | -5.24E-10 | 7.22E-09 | ... | 4.90E-06 | 52458.21741 |
| 2 | 1.621128312 | -6.779826506 | ... | -43171.30534 | -6.56547E+13 |
|  |  |  | ... |  |  |
| 89 | -1.17E-06 | 1.89E-05 | ... | -0.00982473 | 201677489.7 |
| 90 | 8.31E-07 | -1.26E-05 | ... | 0.001542335 | -136918791.3 |

1. *Error layer* 1

Perhitungan yang akan digunakan pada *error layer* 1 ini berdasarkan Persamaan (2.19). Berikut perhitungan *error layer* 1.

= -5.94E-17

= 3.50E-09

= -9.99E-10

= -1.01E-09

= -7.35E-10

= 2.88E-13

= -2.56E-09

= 2.22E-09

= 1.08E-10

= -5.58E-17

= 7.03E-07

= 4.10E-13

Pada perhitungan diatas maka dapat dilihat Tabel 4.41 hasil perhitungan *error* pada *layer* 1 berikut ini.

Tabel 4.41 Error Layer 1

|  |  |  |
| --- | --- | --- |
| Data ke- | Error ke- | Nilai error |
| 1 | 1.1 | -5.94E-17 |
| 1.2 | 3.50E-09 |
| 1.3 | -9.99E-10 |
| 1.4 | -1.01E-09 |
| …. | … |
| 9.9 | 4.10E-13 |
| 2 | 1.1 | 9.20E-11 |
| 1.2 | -1.31E+00 |
| 1.3 | 5.39E-01 |
| 1.4 | 1.81E-01 |
| …. | … |
| 9.9 | -6.87E-06 |
| … | … | … |
| 89 | 1.1 | -8.69E-11 |
| 1.2 | 1.05E-06 |
| 1.3 | -4.45E-08 |
| 1.4 | -1.62E-07 |
| …. | … |
| 9.9 | 3.50E-12 |
| 90 | 1.1 | 5.77E-11 |
| 1.2 | -8.76E-07 |
| 1.3 | 3.53E-08 |
| 1.4 | 1.54E-07 |
| …. | … |
| 9.9 | -2.88E-12 |

1. *Error* parameter dan

Perhitungan parameter *a* dan *c* akan dihitung menggunakan Persamaan (2.24). perhitungan *error* ini adalah sebagai berikut.

Perhitungan *error* parameter *c* akan dihitungan menggunakan Persamaan (2.28). Berikut perhitungan *error* parameter *c*.

Berdasarkan perhitungan diatas, tahap selanjutnya yaitu menentukan perubahan nilai parameter *a* dan *c* menggunakan *gradient descent*. Perhitungan ini akan menggunakan Persamaan (2.32). Berikut perhitungan perubahan nilai parameter *a*.

-1.38E-12

Lalu, perhitungan parameter *c* menggunakan Persamaan (2.33). Berikut perhitungan perubahan paramter *c*.

Kemudian, nilai parameter *a* dan *c* yang telah dihitung sebelumnya akan dijumlahkan dengan nilai parameter *a* dan *c* lama. Pejumlahan itu akan menghasilkan nilai parameter *a* dan *c* yang baru. Perhitungan ini akan menggunakan Persamaan (2.34). Berikut adalah perhitungan nilai parameter *a* baru.

Setelah itu, perhitungan nilai parameter *c* baru adalah sebagai berikut.

Berdasarkan perhitungan nilai parameter *a* dan *c* baru diatas adalah telah dilakukannya perhitungan pada *epoch* 1. Nilai parameter *a* dan *c* dapat dilihat pada Tabel 4.42 berikut ini.

Tabel 4.42 Nilai Parameter dan baru

#### Pengujian ANFIS

Pengujian pada ANFIS ini menggunakan data uji berdasarkan Tabel 4.20 sebagai *input*. Terdapat 5 *layer* yang akan dijalankan pada proses pengujian ini. Pada proses pengujian ini, jaringan ANFIS hanya akan menggunakan tahap *forward* saja. Pada tahap pelatihan yang telah dilakukan sebelumnya akan terbentuk *fuzzy inference system* yang berisi parameter *center* dan *sigma,* fungsi keanggotaan *fuzzy* dan parameter konstan aturan *fuzzy*. Parameter tersebut akan digunakan pada pengujian ANFIS untuk mendapatkan hasil klasifikasi.

1. *Layer* 1

Perhitungan pada *layer* 1 ini akan menggunakan Persamaan (2.30) yang sama digunakan pada tahap pelatihan *layer* 1. Berikut contoh perhitungan pada data ke-1 data uji.

Pada perhitungan diatas maka dapat dilihat Tabel 4.43 hasil perhitunganpada *layer* 1 tahap pengujian berikut ini.

Tabel 4.43 Hasil Perhitungan *Layer* 1 Pengujian ANFIS

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Data ke-** | **Input** | **Mf** | | | | | | | |
| **1** | **2** | **3** | **4** | **...** | **7** | **8** | **9** |
| 1 | 1 | 0.89140 | 0.95477 | 0.73053 | 0.82510 | ... | 0.61038 | 0.09769 | 0.00463 |
|  | 2 | 0.99191 | 0.99817 | 0.99535 | 0.99879 | ... | 0.98338 | 0.99864 | 0.00000 |
|  | 3 | 0.94955 | 0.99509 | 0.96085 | 0.98748 | ... | 0.70334 | 0.86566 | 0.36645 |
|  | 4 | 0.98256 | 0.99228 | 0.99198 | 0.99748 | ... | 0.69432 | 0.99438 | 0.00039 |
|  | 5 | 0.96404 | 0.98469 | 0.91199 | 0.91949 | ... | 0.78291 | 0.15956 | 0.00033 |
|  | 6 | 0.94392 | 0.95613 | 0.81701 | 0.80908 | ... | 0.97628 | 0.01261 | 0.39027 |
|  | 7 | 0.21210 | 0.90803 | 0.10412 | 0.85821 | ... | 0.01649 | 0.07271 | 0.03111 |
|  | 8 | 0.00420 | 0.98634 | 0.81875 | 0.94212 | ... | 0.92313 | 0.53817 | 0.99985 |
|  | 9 | 0.00052 | 0.89091 | 0.99871 | 0.89709 | ... | 0.74733 | 0.27933 | 0.56562 |
| 2 | 1 | 0.96202 | 0.99571 | 0.84045 | 0.71660 | ... | 0.51919 | 0.06190 | 0.00235 |
|  | 2 | 0.99265 | 0.99850 | 0.99592 | 0.99848 | ... | 0.98447 | 0.99832 | 0.00000 |
|  | 3 | 0.98794 | 0.99776 | 0.99323 | 0.94864 | ... | 0.60438 | 0.78097 | 0.28568 |
|  | 4 | 0.96731 | 0.97849 | 0.98189 | 1.00000 | ... | 0.66752 | 0.98455 | 0.00030 |
|  | 5 | 0.98843 | 0.99839 | 0.95277 | 0.87013 | ... | 0.71764 | 0.12635 | 0.00020 |
|  | 6 | 0.96194 | 0.97247 | 0.84857 | 0.77711 | ... | 0.96262 | 0.01048 | 0.42365 |
|  | 7 | 0.00541 | 0.55312 | 0.81095 | 0.13375 | ... | 0.39650 | 0.71458 | 0.50754 |
|  | 8 | 0.04049 | 0.83148 | 0.37075 | 0.91134 | ... | 0.50058 | 0.94473 | 0.72992 |
|  | 9 | 0.04224 | 0.67088 | 0.41619 | 0.66185 | ... | 0.82933 | 0.97554 | 0.95397 |
|  |  |  |  |  |  | ... |  |  |  |
| 10 | 1 | 0.27322 | 0.35682 | 0.15715 | 0.87747 | ... | 0.99056 | 0.59110 | 0.09973 |
|  | 2 | 0.99484 | 0.99940 | 0.99750 | 0.99719 | ... | 0.98770 | 0.99698 | 0.00000 |
|  | 3 | 0.88093 | 0.96131 | 0.89778 | 0.99974 | ... | 0.80577 | 0.93880 | 0.46632 |
|  | 4 | 0.99982 | 0.99904 | 0.99914 | 0.97173 | ... | 0.79047 | 0.99806 | 0.00075 |
|  | 5 | 0.55391 | 0.61139 | 0.46036 | 0.92066 | ... | 0.99321 | 0.54633 | 0.00621 |
|  | 6 | 0.27302 | 0.29106 | 0.16218 | 0.82481 | ... | 0.57475 | 0.24144 | 0.03038 |
|  | 7 | 0.01049 | 0.71521 | 0.68551 | 0.19400 | ... | 0.27485 | 0.58735 | 0.38775 |
|  | 8 | 0.00164 | 0.90840 | 0.93727 | 0.82627 | ... | 0.99190 | 0.38289 | 0.96802 |
|  | 9 | 0.00586 | 0.97950 | 0.81821 | 0.97649 | ... | 0.99688 | 0.65922 | 0.92915 |

1. *Layer* 2

Tahap pengujian pada *layer* 2 ini akan dilakukan perhitungan perkalian derajat keanggotaan yang telah didapat pada *layer* 1. Berikut perhitungan *layer* 2 pada tahap pengujian.

Pada perhitungan diatas maka dapat dilihat Tabel 4.44 hasil perhitunganpada *layer* 2 tahap pengujian berikut ini.

Tabel 4.44 Hasil Perhitungan *Layer* 2 Pengujian ANFIS

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Data ke-** | ***W*** | | | | | | | |
| **1** | **2** | **3** | **4** | **...** | **7** | **8** | **9** |
| **1** | 3.46E-07 | 0.706927873 | 0.043967152 | 0.438014501 | **...** | 0.002549422 | 1.85E-06 | 9.00E-19 |
| **2** | 8.03E-06 | 0.290778083 | 0.082584926 | 0.037028314 | **...** | 0.023448196 | 4.14E-05 | 3.74E-18 |
|  |  |  |  |  |  |  |  |  |
| **10** | 3.66E-09 | 0.038784211 | 0.005519403 | 0.101036513 | **...** | 0.09668162 | 0.010797779 | 1.58E-15 |

1. *Layer* 3

Neuron pada *layer* 3 ini bersifat tetap. Berikut perhitungan pada *layer* 3 ini.

Pada perhitungan diatas maka dapat dilihat Tabel 4.45 hasil perhitunganpada *layer* 3 tahap pengujian berikut ini.

Tabel 4.45 Hasil Perhitungan *Layer* 3 Pengujian ANFIS

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Data ke-** |  | | | | | | | |
| **1** | **2** | **3** | **4** | **...** | **7** | **8** | **9** |
| **1** | 2.87E-07 | 0.586685587 | 0.036488721 | 0.363512042 | **...** | 0.002115787 | 1.53E-06 | 7.47E-19 |
| **2** | 7.75E-06 | 0.280682987 | 0.079717781 | 0.035742782 | **...** | 0.022634132 | 4.00E-05 | 3.61E-18 |
|  |  |  |  |  |  |  |  |  |
| **10** | 1.43E-08 | 0.151535758 | 0.021565138 | 0.394764881 | **...** | 0.377749658 | 0.042188551 | 6.17E-15 |

1. *Layer* 4

Pada *layer* 4 ini dilakukan *defuzzyfication* menggunakan persamaan yang sama pada tahap pelatihan. Berikut perhitungan *layer* 4 pada pengujian ANFIS.

Pada perhitungan diatas maka dapat dilihat Tabel 4.46 hasil perhitunganpada *layer* 4 tahap pengujian berikut ini.

Tabel 4.46 Hasil Perhitungan *Layer* 4 Pengujian ANFIS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data ke- |  | | | | |
| 1 | 2 | ... | 8 | 9 |
| 1 | -8.23E-08 | -2.62393 | ... | 7.11E-07 | 8.71E-19 |
| 2 | -2.78E-06 | -0.60589 | ... | 0.00018 | 4.70E-18 |
|  |  |  | ... |  |  |
| 9 | -1.05E-06 | -2.68326 | ... | -1.95E-06 | 2.38E-21 |
| 10 | -4.50E-09 | -3.67162 | ... | -1.38128 | 2.70E-15 |

1. *Layer* 5

Pada *layer* 5 ini adalah proses penjumlahan terhadap hasil perhitungan pada *layer* 4 sebelumnya. Berikut perhitungan pada *layer* 5 pengujian ANFIS.

Pada perhitungan diatas maka dapat dilihat Tabel 4.47 hasil perhitungan pada *layer* 5 berikut ini.

Tabel 4.47 Hasil Perhitungan *Layer* 5 Pengujian ANFIS

|  |  |  |
| --- | --- | --- |
| **No** | **Hasil Perhitungan** | **Target** |
| 1 | -1.91999 | 0 |
| 2 | 1.96682 | 0 |
| 3 | -1.26757 | 0 |
| 4 | -0.83064 | 0 |
| 5 | 1.27033 | 0 |
| 6 | -0.21501 | 1 |
| 7 | 3.78928 | 1 |
| 8 | 3.58885 | 1 |
| 9 | -2.06106 | 1 |
| 10 | 4.89600 | 1 |

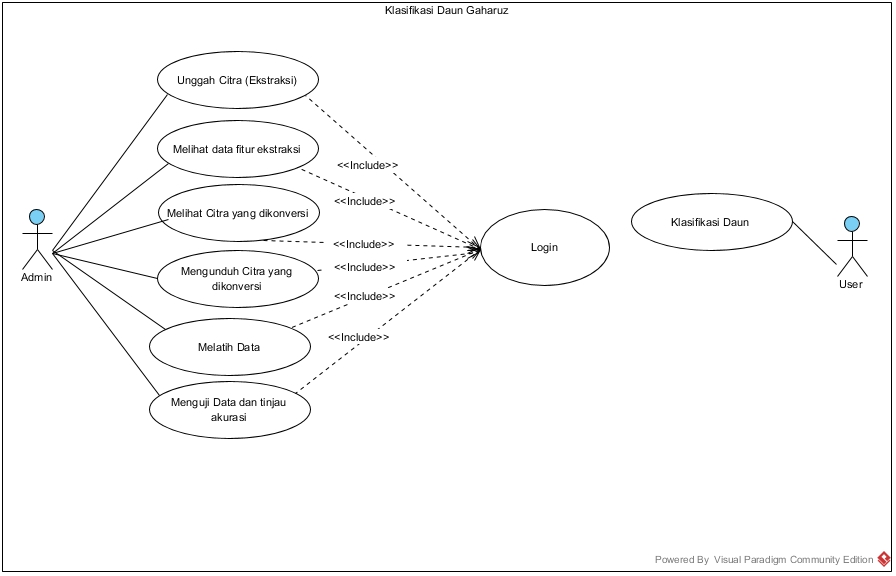
Hasil perhitungan dari *layer* ke-5 pada tahap pengujian ini menunjukan bahwa ada atau tidak hasil klasifikasi yang mencapai target atau cocok dengan target klasifikasinya. Target 0 yaitu *crassna*, target 1 yaitu *microcarpa*, target 2 yaitu *sinensis*, dan target 3 yaitu *subintegra*.

## Analisa Sistem

Analisa sistem dilakukan untuk menyesuaikan kebutuhan dalam menggunakan sistem sebagai pedoman pemodelan. *Unified Modeling Language* (UML) digunakan sebagai pemodelan desain sistem ini. *Use case* diagram, *sequence* diagram, *activity* diagram dan *class* diagram adalah bahasa pemodelan (UML) yang akan diterapkan pada analisa sistem berikut ini.

### *Use case* diagram

Alur dari sistem klasifikasi dan gaharu dapat dilihat pada Gambar 4.15 berikut ini.



Gambar 4.15 *Use case* Diagram

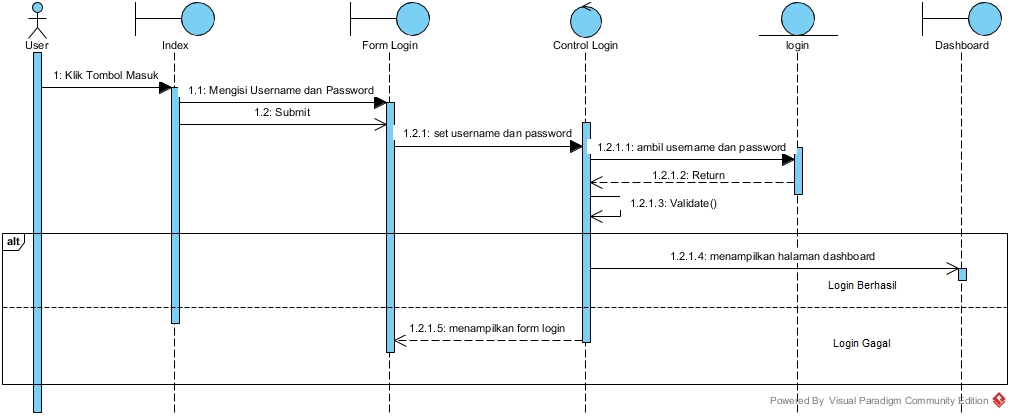
Deskripsi diagram *use* *case* diatas dapat dilihat pada Tabel 4.48 berikut ini.

Tabel 4.48 Deskripsi *Use Case*

|  |  |
| --- | --- |
| **Nama Use Case** | **Deskripsi** |
| Login | Akses ini hanya dapat digunakan oleh Admin sistem |
| Klasifikasi Daun | Klasifikasi daun ini dilakukan tanpa harus login ke sistem, jadi siapapun yang mengakses sistem dapat melakukan klasifikasi citra daun gaharu. |
| Unggah Citra (Ekstraksi) | Unggah citra hanya dapat dilakukan oleh admin untuk menambah data citra baru dan disimpan sebagai model data baru yang telah diekstraksi. |
| Melihat Data Fitur Ekstraksi | Melihat data fitur ekstraksi ini hanya dapat dilakukan oleh admin setelah login. |
| Melihat Citra yang dikonversi | Melihat citra yang dikonversi dengan cara memilih satu data citra yang telah dikonversi. |
| Mengunduh Citra yang dikonversi | Mengunduh nilai citra yang telah dikonversi. Nilai citra yang dapat diunduh yaitu nilai citra *red, green, blue, grayscale* dan biner. |
| Melatih Data | Pelatihan data dilakukan untuk menyimpan model baru. Model baru tersebut digunakan untuk perujukan saat klasifikasi citra. |
| Menguji Data dan Tinjau Akurasi | Pengujian data dilakukan terhadap data latih yang telah disimpan sebagai model dan melihat tingkat akurasi pada pengujian tersebut dalam bentuk persentase. |

### *Sequence* Diagram

*Sequence* diagram akan mengambarkan proses interaksi antara pengguna (*actor*) dengan sistem. Diagram ini didasari oleh *use case* diagram yang telah dibuat sebelumnya. Berikut salah satu *sequence* diagram Gambar 4.16 *sequence* login.

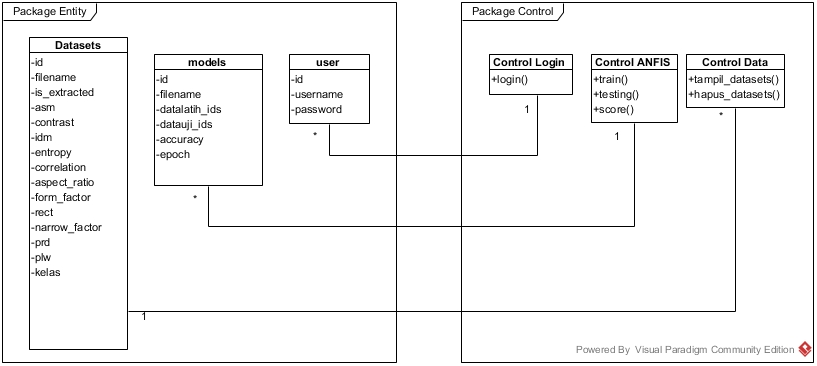


Gambar 4.16 Sequence Login

*Sequence* diagram lainnya dapat dilihat pada bagian **Lampiran A**.

### *Class* Diagram

*Class* diagram ini menjadi acuan pada perancangan *database* sistem*.* Berikut Gambar 4.17 *Class* Diagram Klasifikasi Daun Gaharu.



Gambar 4.17 *Class* Diagaram Klasifikasi Daun Gaharu

## Perancangan

Berdasarkan hasil analisa yang telah dibuat, maka dibuatlah perancangan basis data (*database*) dan perancangan antar muka (*interface*) aplikasi.

### Perancangan basis data (*database*)

Perancangan *database* ini dirancang berdasarkan *class* diagram yang telah dibuat sebelumnya. Berikut Tabel 4.49 Deskripsi tabel *user* pada *database* yang telah dirancang.

Tabel 4.49 Deskripsi Tabel User

|  |  |  |  |
| --- | --- | --- | --- |
| Nama *Field* | Type dan Length | Deskripsi | Keterangan |
| Id | Int | Id User | Primary Key |
| Username | Varchar(20) | Username | Unique |
| Password | Varchar(32) | Password | - |

Tabel useradalah tabel yang digunakan untuk melakukan proses *login* dengan menggunakan *username* dan *password*. Tabel ini memiliki id sebagai *primary key* dengan tipe data *integer* dan *username* yang *unique* dengan tipe data *varchar*. Berikutnya Tabel 4.50 Deskripsi Tabel Models.

Tabel 4.50 Deskripsi Tabel Models

|  |  |  |  |
| --- | --- | --- | --- |
| Nama *Field* | Type dan Length | Deskripsi | Keterangan |
| Id | Int | Id model | Primary key |
| Filename | Text | Nama model | Unique |
| Datalatih\_ids | Text | Penyimpanan nilai model yang telah dilatih | - |
| Datauji\_ids | Text | Penyimpanan nilai akurasi pada pelatihan data | - |
| Accuracy | Real | Penyimpanan nilai Akurasi | - |
| Epoch | Int | Penyimpanan jumlah epoch | - |

Tabel models ini adalah tabel yang berfokus pada proses pelatihan, pengujian dan akurasi data. Id\_model menjadi *primary key* pada tabel ini. Berikutnya Tabel 4.41 Deskripsi Tabel Datasets.

Tabel 4.51 Deskripsi Tabel Datasets

|  |  |  |  |
| --- | --- | --- | --- |
| Nama *Field* | Type dan Length | Deskripsi | Keterangan |
| Id | Int | Id datasets | Primary key |
| Filename | Text | Nama data | - |
| Is\_extracted | varchar | Status ektraksi | - |
| Asm | real | Penyimpanan nilai fitur GLCM | - |
| Contrast | real | Penyimpanan nilai fitur GLCM | - |
| Idm | real | Penyimpanan nilai fitur GLCM | - |
| Entropy | real | Penyimpanan nilai fitur GLCM | - |
| Correlation | real | Penyimpanan nilai fitur GLCM | - |
| Aspect\_ratio | real | Penyimpanan nilai fitur Morfologi | - |
| Form\_factor | real | Penyimpanan nilai fitur Morfologi | - |
| Rect | real | Penyimpanan nilai fitur Morfologi | - |
| Narrow\_factor | real | Penyimpanan nilai fitur Morfologi | - |
| Prd | real | Penyimpanan nilai fitur Morfologi | - |
| Plw | real | Penyimpanan nilai fitur Morfologi | - |
| Kelas | Int | Pembagian kelas pada citra | - |

Tabel datasets ini adalah tabel yang berfungsi sebagai penyimpanan nilai-nilai setiap fitur dan membedakan kelas dari setiap data citra. Id pada tabel ini menjadi *primary key*nya.

### Perancangan Antar Muka (*Interface*) Aplikasi

Perancangan *interface* digunakan sebagai gambaran komunikasi antar pengguna dengan sistem. *Interface* dirancang dengan tampilan yang komunikatif, menarik dan mudah dimengerti oleh pengguna. Berikut Gambar 4.18 adalah rancangan *interface* halaman utama pada sistem klasifikasi daun gaharu.



Gambar 4.18 Rancangan Halaman Utama

Perancangan *interface* lainnya dapat dilihat pada bagian **Lampiran B**.

**DAFTAR PUSTAKA**

Acharya, T., & Ajoy K, R. (2005). *Image Processing: Principles and aplications*. John Wiley & Sons, Inc.

Adi, S., Titi, K., & Erdy, S. (2014). *Panduan Lapangan Pengenalan Jenis Pohon Penghasil Gaharu Aquilaria spp. di Indonesia*.

Budianita, Elvia., Jasril., & Handayani, L. (2015). Implementasi Pengolahan Citra dan Klasifikasi K-Nearest Neighbour Untuk Membangun Aplikasi Pembeda Daging Sapi dan Babi. *Jurnal Sains Dan Teknologi Industri*, *12*, *No 2*(12), 242–247. Retrieved from https://doi.org/10.1139/cjce-2012-0528

Eko, P. (2011). *Pengolahan Citra Digital dan Aplikasinya menggunakan Matlab*. Yogyakarta: Penerbit ANDI.

Fuzzy Yustika, Kana Saputra. (2017). Klasifikasi Belimbing Menggunakan Naive Bayes Berdasarkan Fitur Warna RGB.

Ismail Muhammad, Muhammad Zin Jusoh, A. S. S. (2018). *Gaharu Komoditi Ketiga Negara*. Universiti Putra Malaysia.

Kadir, A. Susanto, A. (2013). *Teori dan Aplikasi Pengolahan Citra*. Yogyakarta: Penerbit ANDI.

Kurt, H. (1989). Multilayer feedforward networks are universal approximators, Neural Networks.

Kusumaewi, S. (2000). *Artificial Intellegentce Teknik dan Aplikasi nya*. Graha Ilmu.

Mega, E. (2018). Aplikasi Pengolahan Citra Untuk Klasifikasi Jenis Buah Mangga.

nur sakinah, retno nugroho, dadan irwan. (2017). Identifikasi buah mangga gedong gincu cirebon berdasarkan citra red-green-blue menggunakan adaptif neuro fuzzy inference system (ANFIS).

Permadi, Y., & Murinto. (2015). Aplikasi pengolahan citra untuk identifikasi kematangan mentimun berdasarkan tekstur kulit buah menggunakan metode ekstraksi ciri statistik. *Jurnal Informatika*, *9*.

Permana, R. A. (2017). Aplikasi klasifikasi jenis – jenis buah jeruk menggunakan metode k-nearest neighbor.

Permenhut. Gaharu termasuk dalam 490 jenis Hasil Hutan Bukan Kayu (HHBK) nabati yang potensial untuk dikembangkan (2007).

Prasetyo, A., & Agustin, S. (2011). Klasifikasi Jenis Pohon Mangga Gadung dan Curut berdasarkan Tekstur Daun. *Sesindo 2011 - Jurusan Sistem Informasi ITS*, *1*.

Ratnasari, E. ginardi R. fathichah. (2017). Klasifikasi penyakit noda pada citra daun tebu berdasarkan ciri tekstur dan warna menggunakan segmented based gray level co-occurence matrix dan lab control moments. *Jurnal Ilmiah Teknologi Sistem Informasi*.

Restu widodo, agus wahyu, arry suprianto. (2018). Pemanfaatan Ciri Gray Leven Co-Occurence Matrix (GLCM) Citra Buah Jeruk Keprok (Citrus Reticulata Blanco) untuk klasifikasi mutu.

Retno Nugroho, Nursinta, S. (2012). Identifikasi Buah Belimbing berdasarkan citra Red-Green Blue menggunakan Adaptive Neuro Fuzzy Inference System (ANFIS).

Rohmi, G. F., Zulfikar, wildan budiawan, & Gerhana, yana aditya. (2018). Implementasi Citra Digital Berdasarkan Nilai HSV Untuk Mengidentifikasi Jenis Tanaman Mangga Menggunakan Algoritma K Nearest Neighbor. *Insight*, *1 No.1*.

Santoso, e & S. (2012). *Budidaya dan Pengembangan Rekayasa Produksi Gaharu*.

Setiawan, H. H. (2018). Implementasi Citra Digital Berdasarkan Nilai HSV Untuk Mengidentifikasi Jenis Tanaman Mangga Menggunakan Algoritma K-Nearest Neighbor.

Shalahuddin M., R. A. t. (2014). *Rekayasa Perangkat Lunak Terstruktur dan Orientasi Objek*. Bandung: Bandung Informatika.

Sutoyo, T. (2009). *Teori Pengolahan Citra Digital*. Yogyakarta: Penerbit ANDI.

Syahid, D. (2016). Sistem Klasifikasi Jenis Tanaman Hias Daun Philodendron Menggunakan Metode K-Nearest Neighboor (KNN) Berdasarkan Nilai Hue, Saturation, Value (HSV).

Syamsuar, F. (2014). PENERAPAN METODE ADAPTIVE NEURO FUZZY INFERENCE SYSTEM (ANFIS) UNTUK PERAMALAN KONSUMSI BBM.

T.C, W. (1989). Canopy Gaps and the Two Major Groups of Forest Trees.

Wiharjo, D., & In, S. (2010). Aplikasi Pengenalan Suara untuk User Interface. *Universitas Bina Nusantara*.

Wijoyono, M. & B. (2002). Anatomi kayu gaharu (Aquilaria Malaccensis Lamk.) dan beberapa jenis sekerabet.