# BAB 2

# LANDASAN TEORI

## 2.1 Pengertian Tanaman Obat

Tanaman obat adalah jenis-jenis tanaman yang memiliki fungsi dan berkhasiat sebagai obat dan dipergunakan untuk penyembuhan maupun mencegah berbagai penyakit. Tanaman obat mengandung zat aktif yang bisa mengobati penyakit tertentu atau jika tidak memiliki kandungan zat aktif tertentu tapi memiliki kandungan efek resultan atau sinergi dari berbagai zat yang mempunyai efek mengobati. Penggunaan tanaman obat sebagai obat bisa dengan cara diminum, ditempel, dan dihirup sehingga kegunaannya dapat memenuhi konsep kerja reseptor sel dalam menerima senyawa kimia atau rangsangan.

Tanaman yang dapat digunakan sebagai obat, baik yang sengaja ditanam maupun tumbuh secara liar. Tumbuhan tersebut digunakan oleh masyarakat untuk diracik dan disajikan sebagai obat guna penyembuhan penyakit. Tanaman obat merupakan salah satu ramuan paling utama produk-produk obat herbal. Tanaman atau bagian tanaman yang diekstraksi dan ekstrak tumbuhan tersebut dipakai sebagai obat.

Berikut jenis tanaman obat yang akan diidentifikasi :

1. Daun Jambu Biji



Gambar 2.1 Daun Jambu

Jambu biji(*Psidium* *guajava*) merupakan salah satu tanaman buah yang banyak ditemukan di wilayah Indonesia, walaupun sebenarnya berasal dari Amerika Tropik. Tanaman ini berbuah sepanjang tahun, sering tumbuh liar, dan umumnya ditemukan pada ketinggian 1-1200 m dpl, serta tumbuh dengan baik pada tanah yang gembur maupun liat. Jambu biji secara taksonomi tergolong ke dalam famili *Myrtaceae*, *genus* *Psidium*, spesies *guajava*.

Daun jambu biji tergolong daun tidak lengkap karena hanya terdiri dari tangkai (petiolus) dan helaian (lamina) saja disebut daun bertangkai. Daun jambu biji memiliki tulang daun yang menyirip (penninervis) yang mana daun ini memiliki satu ibu tulang yang berjalan dari pangkal ke ujung dan merupakan terusan tangkai daun dari ibu tulang kesamping. Pada umumnya warna daun pada sisi atas tampak lebih hijau licin jika di bandingkan dengan sisi bawah karena lapisan atas lebih hijau, jambu biji memiliki permukaan daun yang berkerut (rogosus). Tangkai daun berbentuk silindris dan tidak menebal pada bagian pangkalnya.

Daun jambu biji memiliki banyak kandungan nutrisi seperti asam psidoklat, asam oleanolat, asam guajaverin,vitamin C, antioksidan, flavonoid, serta sifatnya yang anti-inflamasi. Daun jambu biji dikenal sebagai bahan obat tradisional yang memiliki banyak manfaat diantaranya mengobati diare, mengobati maag, menghilangkan kembung, mengatasi jerawat, mengobati sariawan, menurunkan kolesterol, dan meningkatkan metabolisme tubuh.

2. Daun Kersen



Gambar 2.2 Daun Kersen

Tanaman kersen (Muntingia calabura L.) adalah tanaman yang memiliki pertumbuhan yang cepat dan proporsinya ramping. Tanaman ini asli dari Benua Amerika dan banyak dibudidayakan didaerah yang hangat seperti di Asia. Tanaman ini memiliki nama lain: pohon strawberry, cherry jamaican (Inggris), cherry cina atau cherry jepang (India) dan cherry chettu (Telugu). Tanaman kersen merupakan tanaman perdu yang tingginya mencapai 2-10 m dengan daun yang berderet dan dahan menjuntai. Daun kersen memiliki ciri bentuk daun lanset, permukaan bulunya halus, ujung daun runcing, pangkal daun tumpul, tepi daun bergerigi dengan panjang 4–14 cm dan lebar 1–4 cm, daging daun kersen menyerupai kertas dengan tulang daun menyirip dan memiliki warna serupa daunnya namun sedikit kekuningan. Tanaman kersen relatif mudah ditemui, karena tanaman ini dapat tumbuh hampir di semua tempat, seperti di pekarangan rumah, di tepi jalan, bahkan di hutan.

Daun kersen merupakan daun yang kaya manfaatnya untuk kesehatan karena memiliki kandungan alami yang dapat menyehatkan tubuh dan mencegah timbulnya penyakit. Beberapa kandungan yang ada di dalam daun kersen, diantaranya flavonoid (termasuk zat biflavan, flavan, flavanone dan falavone), karbohidrat, alkaloid, tanin, saponin, protein, kalsium, zat besi, lemak tak jenuh ganda, vitamin C, fosfor, zat antioksidan, antiinflamasi, antikanker dan tumor serta zat anti septik. Daun kersen merupakan salah satu dari sekian banyak tanaman yang dapat dimanfaatkan potensinya untuk membantu menjaga kesehatan tubuh. Daun kersen dapat dijadikan minuman yang berkhasiat untuk membantu mengobati diabetes, menyehatkan jantung, mencegah hipertensi, membantu menurunkan kolesterol, mengobati asam urat, dan mengobati batuk.

3. Daun Sirih



Gambar 2.3 Daun Sirih

Tanaman sirih (*Piper* *Betle* *Linn*) berasal dari ordo piperales, famili piperaceae, dan genus piper. Tanaman ini merupakan tanaman yang banyak tersebar di daerah tropis dan subtropis, seperti Sri Lanka, India, Indonesia, Malaysia, Kepulauan Filipina dan Afrika Timur.

Tanaman sirih merambat mencapai ketinggian 15 m dan memiliki daun yang berwarna hijau serta memiliki bentuk seperti hati dengan akar yang merambat. Daun sirih bertekstur lembut pada bagian permukaan dan memiliki ketebalannya sekitar 160-170μm dengan serat trikoma berbentuk silinder menjari. Panjang serat trikomanya kurang lebih 30μm dengan tebal sekitar 5μm. Daunnya memiliki rasa dan bau yang berbeda pada masing-masing daerah di mana ia tumbuh.

Tanaman yang tumbuh menyulur dan memiliki daun lebar ini kaya akan kandungan saponin, tannin, eugenol, dan berbagai jenis minyak esensial. Daun sirih dikenal akan sifat anti-septik, anti-inflamasi, dan pendingin kulit. Khasiatnya tak terbatas di permukaan tubuh. Kandungan-kandungan tersebut menjadikan daun sirih mempunyai manfaat bagi kesehatan tubuh seperti mengatasi mimisan, menghilangkan bau mulut, menghilangkan kotoran pada mata, mengobati nyeri, membantu proses penyembuhan luka, menghambat karies gigi, mengatasi gangguan pencernaan, dan sebagai antiseptik.

## 2.2 Pengolahan Citra (*Image Processing*)

Pengolahan citra merupakan sebuah ilmu dalam kecerdasan buatan. Pengolahan citra dapat juga definiskan sebagai salah satu cabang ilmu dari kecerdasan buatan yang mempelajari tentang bangaimana suatu pola dapat dikenali oleh sebuah mesin atau komputer, lalu mesin tersebut dapat menentukan citra yang sudah dikenali dalam suatu kelas. Pengolahan citra memiliki 3 tahap penting yaitu *preprocessing*, *feature* *extraction* atau ekstraksi fitur, dan klasifikasi.

### 2.2.1 *Preprocessing*

*Preprocessing* adalah tahap awal pada proses pengenalan pola yang berguna untuk memisahkan citra atau gambar dengan latar belakang. Hal yang dilakukan pada *preprocessing* yaitu mengubah suatu citra menjadi citra *grayscale*. Hal lain yang dilakukan setelah *grayscaling* adalah merubah ukuran suatu citra menjadi *pixel* (*resize*) yang lebih kecil agar dapat terlihat pola dari suatu citra.

### 2.2.2 *Feature Extraction*

*Feature Extraction* atau ekstraksi fitur adalah tahap kedua dalam proses pegenalan pola yang bertujuan untuk memperoleh informasi yang lebih jelas mengenai data dalam sebuah citra. Hal yang dilakukan pada tahap ekstraksi fitur adalah merubah citra menjadi bentuk pola agar dapat dikenali oleh sistem. Metode ekstraksi fitur pada pengenalan pola beragam jenisnya, diantaranya adalah Transformasi *Wavelet*, *Gray Level Co-Occurence Matrix* (GLCM), *Principal Component Analysis* (PCA) dan deteksi tepi *Canny*. Penelitian ini menggunakan metode ekstraksi fitur GLCM (*Gray Level Co-Occurence Matrix*). GLCM adalah matriks yang menggambarkan frekuensi munculnya pasangan dua piksel dengan intensitas tertentu dalam jarak d dan orientasi arah dengan sudut θ tertentu dalam citra.

### 2.2.3 Klasifikasi

Klasifikasi merupakan suatu metode untuk mengelompokkan sebuah objek ke dalam kelompok atau kelas tertentu. Algoritma klasifikasi yang banyak digunakan secara luas, yaitu *Decision* atau *Classification* *Trees*, *Bayesian* *Classifiers* atau *Naïve* *Bayes* *classifiers*, *Neural* *Networks*, Analisa Statistik, Algoritma Genetika, *Rough* *Sets*, *K-Nearest Neighbor*, Metode *Rule* *Based*, *Memory* *Based* *Reasoning*, dan *Support* *Vector* *Machines* (SVM). Penelitian ini menggunakan metode klasifikasi *K-Nearest Neighbor*. Proses ini dilakukan agar data atau citra dapat dikategorikan dalam suatu kelas tertentu yang telah ditentukan.

### 2.3 CIELab

Warna adalah deskriptor kuat dalam segmentasi citra yang menyederhanakan identifikasi objek dan ekstraksi dari gambar. Model warna memfasilitasi spesifikasi warna dengan cara yang standar. Sebuah sub ruang sebuah model warna memberikan satu titik untuk mewakili warna.

CIELab adalah salah satu struktur warna yang didefinisikan CIE (Commicion International de 1’Eclairage). Pada CIELab, besaran CIE\_L\* untuk mendeskripsikan kecerahan warna, 0 untuk hitam dan L\* = 100 untuk putih. Dimensi CIE\_a\* mendeskripsikan jenis warna hijau – merah, dimana angka negatif a\* mengindikasikan warna hijau dan sebaliknya CIE\_a\* positif mengindikasikan warna merah. Dimensi CIE\_b\* untuk jenis warna biru – kuning, dimana angka negatif b\* mengindikasikan warna biru dan sebaliknya CIE\_b\* positif mengindikasikan warna kuning.

### 2.4 Segmentasi Citra

Segmentasi merupakan teknik awal dalam pengolahan citra yang berfungsi untuk membagi bagian-bagian yang sama. Segmentasi adalah teknik awal yang digunakan untuk mengubah citra masukan ke dalam keluaran berdasarkan properti yang diambil dari citra tersebut. Segmentasi berfungsi membagi bagian-bagian citra ke dalam objek intesitasnya masing masing sehingga bisa dibedakan antara objek (*foreground*) dengan *background*. Pembagian ini tergantung pada citra uji yang akan digunakan. Algoritma dari segmentasi terbagi dalam dua macam. Diskontinuitas adalah segmentasi pada citra berdasarkan perbedaan dalam intesitasnya, contoh titik, garis dan tepi (edge). Similaritas adalah segmentasi berdasarkan persamaan kriteria atribut yang dimiliki citra tersebut, contohnya t*hresholding*, *region growing*, *region splitting*, dan *region merging*.

Pada penelitian ini menggunakan jenis segmentasi jenis pengambangan (*thresholding*). *Thresholding* merupakan salah satu jenis teknik segmentasi citra dimana prosesnya didasarkan pada perbedaan skala keabuan pada sebuah citra. Pada tahapan ini bertujuan menghasilkan citra biner yaitu hitam dan putih.

### 2.5 Operasi Morfologi

Operasi morfologi citra merupakan suatu proses yang bertujuan untuk mengubah bentuk objek pada citra asli. Proses tersebut dapat dilakukan pada citra *grayscale* maupun citra biner. Jenis-jenis operasi morfologi di antaranya adalah dilasi, erosi, *closing*, dan *opening*. Pada penelitian ini menggunakan operasi morfologi *filling* *holes*, operasi ini bertujuan bertujuan untuk mengisi keseluruhan region menjadi 1. Operasi ini menggunakan acuan berdasarkan nilai piksel tetangganya. Gambar 2.4 menjukkan proses filling holes.

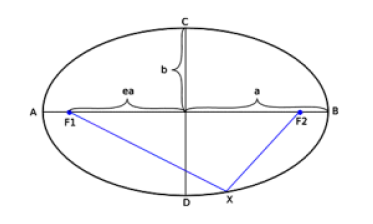


Gambar 2.4 Proses filling holes

Dari Gambar 2.4 dapat dilihat secara kasat mata bahwa citra awal memiliki sebuah lubang (*hole*) dapat dihilangkan dengan operasi *filling* *holes*. Citra masukkan adalah citra biner yang memiliki lubang (*hole*), kemudian dilakukan pengisian sehingga mendapatkan obyek yang maksimal.

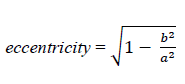
### 2.6 Ekstraksi Ciri Bentuk

Bentuk dapat didefinisikan sebagai gambaran dari suatu objek dalam posisi, orientasi dan ukuran. Ciri bentuk dalam suatu citra sangat esensial untuk segmentasi citra karena dapat mendeteksi objek atau batas wilayah. Untuk membedakan bentuk objek satu dengan objek lainnya, dapat menggunakan parameter yang disebut dengan *eccentricity*. *Eccentricity* merupakan nilai perbandingan antara jarak foci ellips minor dengan foci ellips mayor suatu objek. *Eccentricity* memiliki rentang nilai antara 0 hingga 1. Objek yang berbentuk memanjang atau mendekati bentuk garis lurus maka nilai *eccentricity* mendekati angka 1, sedangkan objek yang berbentuk bulat atau lingkaran maka nilai *eccentricity* mendekati angka 0.

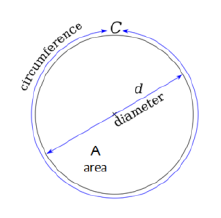


Gambar 2.5 Penghitungan *Eccentricity*

Perhitungan *eccentricity* dihitung dengan cara seperti berikut :

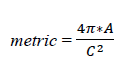


Dimana a menunjukkan major axis (jari-jari terpanjang) dari suatu citra dengan ukuran piksel, sedangkan b menunjukkan minor axis (jari-jari terpendek) dari suatu citra dengan ukuran piksel. Parameter lainnya yang dapat digunakan untuk membedakan bentuk suatu objek yaitu *metric*. *Metric* merupakan nilai perbandingan antara luas dan keliling objek yang disebut juga dengan ukuran kebulatan (*roundness*). *Metric* memiliki rentang nilai antara 0 hingga 1. Objek yang berbentuk memanjang atau mendekati bentuk garis lurus, nilai *metric* mendekati angka 0, sedangkan objek yang berbentuk bulat atau lingkaran, nilai *metric* mendekati angka 1.



Gambar 2.6 Penghitungan *Metric*

Perhitungan *metric* dihitung dengan cara seperti berikut :



Dengan C adalah keliling sebuah citra dengan ukuran piksel, dan A adalah luas wilayah citra dengan satuan piksel. Untuk membedakan ukuran objek satu dengan objek lainnya dapat menggunakan parameter luas (area) dan keliling (perimeter). Luas yang disebut juga dengan area merupakan banyaknya piksel yang menyusun suatu objek. Sedangkan keliling yang disebut juga dengan perimeter merupakan banyaknya piksel yang mengelilingi suatu objek. Berdasarkan dari ukuran nilai *metric*, area, dan perimeter dapat diturunkan sebuah nilai yang disebut dengan *shape factor* misalnya nilai *roundness*, *aspect* *ratio*, dan *triangle*.

### 2.7 Ekstraksi Ciri Tekstur

Tekstur merupakan ciri citra yang dapat digunakan untuk menentukan karakterisasi suatu citra karena tekstur mengandung informasi penting mengenai susunan struktur permukaan suatu citra . Untuk membedakan tekstur objek satu dengan objek lainnya dapat menggunakan ciri statistik orde dua. Ciri orde dua didasarkan pada probabilitas hubungan ketetanggaan antara dua piksel pada jarak dan orientasi sudut tertentu. Salah satu metode yang dapat menghasilkan ciri orde dua adalah *Gray Level Co-occurrence Matrix* (GLCM).

*Gray Level Co-occurrence Matrix* (GLCM) diusulkan pertama kali oleh Haralick pada tahun 1973 dam memiliki 28 fitur untuk menjelaskan pola spasial. Misalkan, f(x,y) adalah citra dengan ukuran Nx dan Ny yang memiliki piksel dengan kemungkinan hingga L level dan vektor r adalah vektor arah ofset spasial. GLCM→(i,j) didefinisikan sebagai 𝑟 jumlah piksel dengan j1, ..., L yang terjadi pada ofset vektor r terhadap piksel dengan nilai i1, ..., L, yang dapat dinyatakan dalam rumus :



Sebagai ilustrasi, ketetanggaan piksel dapat dipilih ke arah timur(kanan). Salah satu cara untuk merepresentasikan hubungan ini yaitu berupa (1,0), yang menyatakan hubungan dua piksel yang sejajar secara horizontal dengan piksel bernilai 1 diikuti dengan piksel bernilai 0, sehingga jumlah kelompok piksel yang memenuhi hubungan tersebut dihitung.



Gambar 2.7 Contoh Penentuan Awal Matriks GLCM

Matriks pada Gambar 2.7 dinamakan *matrix framework*. Matriks ini kemudian diolah menjadi matriks simetris dengan cara menambahkan dengan hasil tranposnya, seperti yang terlihat pada Gambar 2.8.



Gambar 2.8 Matriks *framework* menjadi matriks simetris

Untuk menghilangkan ketergantungan pada ukuran citra, nilai-nilai elemen GLCM perlu dinormalisasi sehingga jumlahnya bernilai 1. Dengan demikian, hasil normalisasi dari matriks GLCM pada Gambar 2.7.



Gambar 2.9 Normalisasi matriks GLCM

Untuk mendapatkan fitur tekstur GLCM, hanya 14 besaran yang diusulkan oleh Haralick untuk dipakai. Beberapa fitur yang akan dipakai nantinya adalah *contrast, correlation, energy,* dan *homogeneity*.

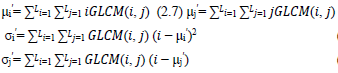
1. *Contrast* merupakan ukuran keberadaan variasi aras keabuan piksel citra dihitung dengan cara seperti berikut:



1. *Correlation* merupakan ukuran ketergantungan linear antar nilai aras keabuan dalam citra. *Correlation* dihitung dengan cara seperti berikut:



Dengan :



1. *Energy* merupakan nilai dari jumlah kuadrat pada elemen-elemen matriks GLCM. *Energy* dihitung dengan cara seperti berikut:



1. *Homogeneity* digunakan untuk mengukur homogenitas yaitu ukuran kedekatan distribusi masing-masing elemen pada matriks GLCM ke matriks GLCM diagonal. *Homogeneity* dihitung dengan cara persamaan 2.8 berikut:



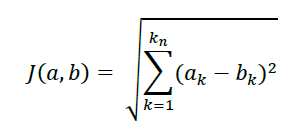
### 2.8 *K-Nearest Neighbor* (KNN)

KNN merupakan salah satu algoritma pembelajaran mesin sederhana. Hal ini hanya didasarkan pada gagasan bahwa suatu objek yang dekat satu sama lain juga akan memiliki karakteristik yang mirip. Ini berarti jika kita mengetahui ciri-ciri dari salah satu objek, maka kita juga dapat memprediksi objek lain berdasarkan tetangga terdekatnya. KNN adalah improvisasi lanjutan dari teknik klasifikasi *Nearest Neighbor*. Hal ini didasarkan pada gagasan bahwa setiap contoh baru dapat diklasifikasikan oleh suara mayoritas dari k tetangga, di mana k adalah bilangan bulat positif, dan biasanya dengan jumlah kecil. Algoritma klasifikasi KNN memprediksi kategori tes sampel sesuai dengan sampel pelatihan k yang merupakan tetangga terdekat dengan sampel uji, dan memasukkan ke dalam kategori yang memiliki kategori probabilitas terbesar.

Dalam pengenalan pola, algoritma KNN adalah metode yang digunakan untuk mengklasifikasikan objek berdasarkan contoh pelatihan terdekat di ruang fitur. KNN adalah jenis *insctance-based learning*, atau *lazy* *learning* dimana fungsi ini hanya didekati secara lokal dan semua perhitungan ditangguhkan sampai klasifikasi.

Metode klasifikasi KNN memiliki beberapa tahap, yang pertama nilai k yang merupakan jumlah tetangga terdekat yang akan menentukan *query* baru masuk ke kelas mana ditentukan. Tahap kedua, k tetangga terdekat dicari dengan cara menghitung jarak titik *query* dengan titik training. Tahap ketiga, setelah mengetahui jarak masing-masing titik training dengan titik *query*, kemudian lihat nilai yang paling kecil. Tahap keempat ambil k nilai terkecil selanjutnya lihat kelasnya. Kelas yang paling banyak merupakan kelas dari *queri* baru.

Dekat atau jauhnya jarak titik dengan tetangganya bisa dihitung dengan menggunakan *Euclidean* *Distance*. *Euclidean* *distance* direpresentasikan sebagai berikut :



Dengan :

J(a,b) = jarak *Euclidean* antara vector a dan vektor b

ak = komponen ke k dari vektor k

bk = komponen ke k dari vektor k

kn = jumlah komponen pada vektor a dan vektor b

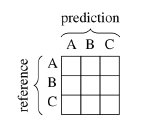
J(a,b) merupakan jarak antara titik a yang merupakan titik yang telah diketahui kelasnya dan b berupa titik baru. Jarak antara titik baru dengan titik-titik training dihitung dan diambil k buah titik terdekat. Titik baru diprediksi masuk ke kelas dengan klasifikasi terbanyak dari titik-titik tersebut.

**2.9 Parameter Evaluasi**

Sebagai indikator tingkat performansi akurasi dari pencocokan citra uji terhadap citra yang telah ada dalam database, dalam penelitian ini digunakan beberapa parameter yang berfungsi untuk memudahkan analisis dari performa pengenalan. Indikator yang digunakan yaitu *true* *positive* (TP), *true* *negative* (TN), *false* *positive* (FP) dan *false* *negative* (FN) yang merupakan elemen dari confusion matrix.

1. *True* *Positive* (TP) adalah jumlah data positif yang benar diklasifikasi oleh *classifier*.
2. *True Negative* (TN) adalah jumlah data negatif yang benar diklasifikasi oleh *classifier*.
3. *False Positive* (FP) adalah jumlah data negatif yang salah diklasifikasi sebagai data positif.
4. *False Negative* (FN) adalah jumlah data positif yang salah diklasifikasi sebagai data negatif.

*Recall*, *precision* dan *F-Measure* merupakan parameter yang dgunakan untuk mengukur kefektifan dan mengevaluasi informasi temu kembali. *Recall* menyatakan perbandingan jumlah *true* *positive* terhadap penjumlahan antara *true* *positive* dan *false* *negative*. Sedangkan *precision* menyatakan perbandingan perbandingan jumlah *true* *positive* terhadap penjumlahan antara *true* *positive* dan *false* *positive*. Penghitungan *recall* dan *precision* dalam penelitian ini terbentuk dari tiga elemen yaitu citra daun jambu biji, kersen, dan daun sirih dimana ketiganya digunakan untuk membentuk *confusion* *matrix*. Penghitungan *recall*, *precision* dan *F-Measure* dari confusion matrix yang terdiri dari tiga elemen disajikan seperti pada Gambar 2.10



Gambar 2.10 *Confusion* *matrix* 3x3

Berdasarkan Gambar 2.10, *recall* didapatkan dengan cara membagi nilai satu elemen matrix terhadap keseluruhan elemen yang ada pada baris matrix dimana elemen tersebut berada.

*Recall* =

*Precision* didapatkan dengan cara membagi nilai satu elemen matrix terhadap keseluruhan elemen yang ada pada kolom matrix dimana elemen tersebut berada.

*Precision* =

Sedangkan *F-Measure* informasi temu kembali yang mengkombinasikan recall dan precision.

*F-Measure* =

### 2.10 Penelitian Terdahulu

Penelitian dalam mengidentifikasi jenis daun sudah pernah dilakukan dengan menggunakan metode support vector machine. Analisis tekstur yang digunakan adalah Gray Level Cooccurrences Matrix (GLCM) dengan parameter sudut 0° dan jarak 1 piksel dan tujuh ekstraksi ciri dari GLCM yang dilakukan antara lain energy, contrast, entropy, dissimilarity, homogeneity, correlation dan maximum probability. Sedangkan analisis warna yang digunakan adalah *color moment* mengekstraksi ciri warna dari *moment* *mean* dan standar deviasi dari masing – masing *channel* (HSV), sehingga menghasilkan enam fitur warna yaitu tiga dari *moment* *mean* (HSV) dan tiga dari *moment* standar deviasi (HSV). Data latih dari penelitian ini menggunakan 100 gambar perjenis kategori dan 30 gambar perjenis kategori sebagai data uji. Tingkat akurasi yang didapatkan sebesar 87% dari tiga jenis kategori (Puji et al., 2018).

Setelah itu ada juga penelitian tentang identifikasi identifikasi jenis tumbuhan menggunakan jaringan syaraf tiruan *backpropogation*. Pada penelitian ini menggunakan 4 jenis nama daun seperti daun bougenvillea, daun geranium, daun magnolia soulangeana, dan daun pinus dengan 16 sampel citra daun dengan bentuk daun yang berbeda-beda untuk setiap jenisnya. Hasil pengujian yang didapatkan menunjukan tingkat akurasi sebesar 93,6% (Reni et al,. 2018).

GLCM juga diaplikasikan dalam penelitian identifikasi tumbuhan obat herbal. GLCM digunakan sebagai analisis tekstur dengan mengekstrak nilai kontras, korelasi, energi dan homogenitas serta klasifikasi dilakukan dengan K-Nearest Neighbor (KNN). Ada 10 kategori tumbuhan obat herbal yang digunakan dengan 9 gambar perjenis kategori sebagai data latih. Tingkat akurasi yang didapatkan sebesar 83,33% (Fittria et al., 2018).

Penelitan klasifikasi tanaman obat dengan menggunakan metode jaringan syaraf tiruan *backpropogation* berdasarkan fitur morfologipernah dilakukan oleh Kana Saputra S. Dataset yang digunakan pada penelitian ini yaitu 60 data latih yang dibagi menjadi 5 kategori dan 15 data uji. Hasil *recognition* *rate* yang didapatkan sebesar 90% untuk data training dan 75,56% untuk data testing (Kana Saputra S, 2018).

Penelitian tanaman buah juga pernah dilakukan dengan menggunakan algoritma *Learning* *Vector* *Quantization* (LVQ). Ekstraksi fitur yang digunakan berdasarkan tekstur, warna dan bentuk. Terdapat 6 paramater yang digunakan dalam ekstraksi fitur bentuk yaitu *slimness*, *roundness*, *rectangularity*, *narrow* *factor*, rasio keliling dan diameter, rasio perimeter dengan panjang dan lebar. Ekstraksi fitur warna terdiri dari *mean*, *skewness*, dan *kurtosis*. Sedangkan Ekstraksi fitur bentuk yang digunakan yaitu *contrast*, *correlation*, *energy*, dan *homogeneity*, *maximum* *probability* dan *entropy*. Tingkat akurasi keberhasilan sistem mengenali daun mencapai 82% dengan menggunakan 50 data sampel daun sebagai data latih (Sutarno et al., 2017).

Tabel 2.1 Penelitian Terdahulu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Peneliti** | **Judul Penelitian** | **Metode** | **Tahun** |
| 1 | Puji et al | Klasifikasi Penyakit Daun Kentang Berdasarkan Fitur Tekstur Dan Fitur Warna Menggunakan Support Vector Machine | Support Vector Machine | 2018 |
| 2 | Reni et al | Identifikasi Jenis Tumbuhan Menggunakan Citra Daun Berbasis Jaringan Saraf Tiruan (Artificial Neural Network) | Artificial Neural Network | 2018 |
| 3 | Fittria et al | Identifikasi Tumbuhan Obat Herbal Berdasarkan Citra Daun Menggunakan  Algoritma Gray Level Co-Occurence Matrix Dan K-Nearest Neighbor | K-Nearest Neighbor | 2018 |
| 4 | Kana Saputra S | Klasifikasi Tanaman Obat Berdasarkan Ekstraksi Fitur Morfologi Daun Menggunakan Jaringan Syaraf Tiruan | Artificial Neural Network | 2018 |
| 5 | Sutarno et al | Identifikasi Tanaman Buah Berdasarkan Fitur Bentuk, Warna, Dan Tekstur Daun Berbasis Pengolahan Citra Dan *Learning* *Vector* *Quantization* (LVQ) | Learning Vector Quantization | 2017 |