LAPORAN PROYEK AKHIR MATA KULIAH DATA MINING

PENERAPAN HSV, GLCM, PCA DAN KNN UNTUK KLASIFIKASI CITRA TANAH DALAM MENENTUKAN KESESUAIAN TANAMAN

OLEH:

535230024	Shirley
535230026	Claresta Elysia
535230031	Heru Purnomo

PROGRAM STUDI SARJANA TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS TARUMANAGARA

JUNI 2025

DAFTAR ISI

JUDUL **0**
DAFTAR ISI **1**
DAFTAR GAMBAR **3**
DAFTAR TABEL **4**
BAB I PENDAHULUAN **5**
1.1 Latar belakang 5
1.2 Maksud dan Tujuan Penelitian 7
1.3 Batasan Masalah 8
BAB II LANDASAN TEORITIK **10**
2.1 Citra Digital 10
2.2 MATLAB 11
2.3 RGB 12
2.4 *Hue Staturation Value* 13
2.4.1 *Hue* (Nuansa Warna) 14
2.4.2 *Saturation* (Kemurnian Warna) 14
2.4.3 *Value* (Tingkah Kecerahan) 15
2.4.4 Rumus HSV 15
2.5 *Gray Level Co-Occurrence Matrix* 17
2.5.1 Kontras (Contrast) 18
2.5.2 Homogenitas (*Homogenity*) 19
2.5.3 *Energy* (ASM) 19
2.5.4 Korelasi (*Correlation*) 20
2.6 *Principal Component Analysis* 21
2.6.1 Normalisasi Data 22
2.6.2 Kalkulasi Matriks Kovarians 22
2.6.3 Menghitung Nilai dan Vektor Eigen 23
2.6.4 Proyeksi ke Dimensi Baru 24
2.7 *K-Nearest Neighbor* 25
2.7.1 Menghitung Jarak *Euclidean* 25
2.7.2 Menentukan k Tetangga Tedekat 26
2.7.3 Menentukan Kelas Mayoritas 27

BAB III METODE EKSPERIMEN 29
3.1 Pengumpulan Data.....29
3.1.1 Black Soil.....30
3.1.2 Cinder Soil 31
3.1.3 Laterite Soil.....32
3.1.4 Peat Soil 34
3.1.5 Yellow Soil 35
3.2 Pra-Pemrosesan 36
3.3 Ekstraksi Warna HSV 41
3.4 Ekstraksi GLCM 43
3.5 Reduksi PCA.....45
3.6 Klasifikasi 48
3.7 Diagram Alir.....50
3.7.1 Pengumpulan Dataset dari Kaggle.....51
3.7.2 Pra-pemrosesan Data.....51
3.7.3 Ekstraksi Fitur Warna.....52
3.7.4 Ekstraksi Fitur Tekstur.....52
3.7.5 Reduksi.....52
3.7.6 Klasifikasi.....53
BAB IV HASIL DAN PEMBAHASAN.....54
4.1 Pelatihan dan Pengujian Eksperimen 54
4.2 Hasil Eksperimen 57
4.3 Analisis dan Pembahasan 61
BAB V KESIMPULAN DAN SARAN.....65
5.1 Kesimpulan 65
5.2 Saran 66
5.2.1 Pengembangan Sistem Terintegrasi dengan Database Kesesuaian Tanaman 66
5.2.2 Pengembangan Antarmuka Pengguna (UI) dan Pengalaman Pengguna (UX) yang Intuitif.....68
DAFTAR PUSTAKA 70
LAMPIRAN.....73

DAFTAR GAMBAR

Gambar 3.1. 1 Dataset citra tanah sebelum sotir 30

Gambar 3.1.1. 1 Tampilan beberapa dataset black soil sebelum pre-processing 31

Gambar 3.1.2. 1 Tampilan beberapa dataset cinder soil sebelum pre-processing 32

Gambar 3.1.3. 1 Tampilan beberapa dataset laterite soil sebelum pre-processing 33

Gambar 3.1.4. 1 Tampilan beberapa dataset peat soil sebelum pre-processing 35

Gambar 3.1.5. 1 Tampilan beberapa dataset yellow soil sebelum pre-processing 36

Gambar 3.2. 1 Proses Cropping salah satu Gambar.....37

Gambar 3.2. 2 Hasil Resize 224 x 224 piksel black soil 38

Gambar 3.2. 3 Hasil Resize 224 x 224 piksel laterite soil.....38

Gambar 3.2. 4 Hasil Resize 224 x 224 piksel peat soil 39

Gambar 3.2. 5 Hasil Resize 224 x 224 piksel yellow soil 39

Gambar 3.7. 1 Diagram Alir Proses Eksperimen 51

Gambar 4.2.2. 1 Scatter PCA 3D 58

Gambar 4.2.2. Scatter PCA 3D (Tampak Atas) 59

Gambar 4.2.3. 1 Confusion Matrix 60

DAFTAR TABEL

Tabel 3.1.1. 1 Karakteristik Black Soil 31

Tabel 3.1.2. 1 Karakteristik Cinder Soil 32

Tabel 3.1.3. Karakteristik Laterite Soil 33

Tabel 3.1.4. 1 Karakteristik Peat Soil 34

Table 3.2. 1 Pembagian Data Pelatihan dan Data Pengujian 40

BAB I

PENDAHULUAN

1.1 Latar belakang

Indonesia merupakan negara yang memiliki sumber daya alam terbanyak di dunia [1]. Indonesia memiliki potensi besar dalam sektor pertanian dengan tanah yang subur. Sektor pertanian Indonesia memberikan kontribusi signifikan terhadap perekonomian negara, dengan pertumbuhan produksi sebesar 1,48% dan kontribusi terhadap Produk Domestik Bruto (PDB) mencapai 13,57% [2]. Selain itu, sektor ini juga berperan sebagai penopang utama kehidupan masyarakat, penyerap tenaga kerja terbesar, serta penyumbang devisa negara. Oleh karena itu, pengembangan sektor pertanian menjadi prioritas penting untuk membangun ketahanan ekonomi nasional.

Namun, seiring meningkatnya tantangan di tingkat global dan kebutuhan akan sistem pertanian yang lebih efisien, dibutuhankan cara yang lebih praktis dan akurat dalam mengelola serta memanfaatkan sumber daya alam. Salah satu aspek penting dalam hal ini adalah menentukan kesesuaian tanah dengan jenis tanaman tertentu. Untuk mendukung hal tersebut, diperlukan metode yang mampu mengklasifikasikan jenis tanah secara efektif dan efisien.

Dalam upaya menjawab kebutuhan tersebut, penelitian ini menggabungkan beberapa metode komputasi dalam proses klasifikasi

citra tanah, yaitu *Hue Saturation Value* (HSV), *Gray Level Co-occurrence Matrix* (GLCM), *Pincipal Component Analysis* (PCA), dan *K-Nearest Neighbor* (KNN). Pendekatan ini bertujuan untuk menggabungkan fitur warna dan tekstur dari citra tanah, yang kemudian diproses melalui teknik reduksi dimensi dan algoritma klasifikasi guna menghasilkan sistem klasifikasi yang akurat.

Metode HSV digunakan untuk mengekstraksi fitur warna dengan mengubah citra dari model warna RGB (*Red, Green, Blue*) ke model HSV (*Hue, Saturation, Value*). HSV dipilih karena mampu merepresentasikan warna yang mendekat presepsi manusia atau warna yang dapat ditangkap oleh retina mata [3]. Selanjutnya, Metode GLCM digunakan untuk mengekstraksi fitur tekstur citra berdasarkan hubungan antara piksel pada tingkat keabuan, dengan mempertimbangkan jarak dan sudut tertentu [4].

Setelah fitur warna dan tekstur diperoleh, metode PCA digunakan untuk mereduksi jumlah fitur tersebut agar proses klasifikasi menjadi lebih cepat, tanpa menghilangkan informasi penting. Terakhir, metode KNN adalah metode non-parametrik yang sederhana untuk melakukan klasifikasi. Meskipun algoritma KNN tergolong sederhana, tetapi performanya sangat baik dan sering digunakan sebagai metode pembanding dalam berbagai penelitian [5].

Dengan menggabungkan fitur warna dari HSV, fitur tekstur dari GLCM, dan pemrosesan data menggunakan PCA dan KNN, sistem ini diharapkan mampu mengklasifikasikan jenis tanah dengan lebih akurat dan efisien di berbagai kondisi lingkungan.

1.2 Maksud dan Tujuan Penelitian

Penelitian ini dimaksudkan untuk mengembangkan sistem klasifikasi jenis tanah yang dapat membantu dalam menentukan kesesuaian lahan untuk berbagai jenis tanaman berdasarkan karakteristik citra tanah. Sistem ini diharapkan dapat memberikan informasi yang akurat mengenai jenis tanah sehingga mempermudah proses pengambilan keputusan dalam memilih tanaman yang sesuai dengan kondisi lahan.

Tujuan dari penelitian ini adalah untuk mengembangkan pendekatan klasifikasi jenis tanah menggunakan teknik pengolahan citra digital. Pendekatan ini melibatkan beberapa metode, yaitu ekstraksi fitur warna menggunakan *Hue Saturation Value* (HSV), dan fitur tekstur menggunakan *Gray Level Co-occurrence Matrix* (GLCM). Setelah fitur diperoleh, metode *Principal Component Analysis* (PCA) digunakan untuk mereduksi dimensi data tanpa menghilangkan informasi penting. Kemudian, proses klasifikasi jenis tanah dilakukan menggunakan algoritma *K-Nearest Neighbor* (KNN). Dengan menggabungkan metode-metode tersebut, diharapkan dapat

menghasilkan sistem klasifikasi tanah yang akurat dan efisien, sehingga dapat digunakan sebagai dasar dalam menentukan jenis tanaman yang sesuai dengan karakteristik tanah.

1.3 Batasan Masalah

Penelitian ini dibatasi pada penggunaan data citra tanah yang hanya mencakup empat dari lima jenis tanah yang tersedia dalam dataset, yaitu *Black Soil*, *Laterite Soil*, *Peat Soil*, dan *Yellow Soil*. Jenis tanah *Cinder Soil* tidak digunakan karena kurang relavan dan jarang ditemukan di wilayah Indonesia.

Selain itu, jumlah gambar yang digunakan dalam penelitian ini juga terbatas. Dari total 156 gambar dalam dataset asli, hanya 120 gambar yang dipilih setelah melalui proses seleksi kualitas dan relavansi. Beberapa gambar dihapus karena kualitasnya rendah atau tidak sesuai dengan tujuan penelitian.

Gambar-gambar yang digunakan telah melalui proses *crop* untuk memfokuskan area tanah dan menghilangkan bagian luar gambar yang tidak diperlukan. Karena jumlah data masih terbatas, dilakukan proses argumentasi data menggunakan teknik rotasi serta flip horizontal dan vertikal. Hasil augmentasi ini menghasilkan total 429 gambar yang digunakan dalam proses pelatihan dan pengujian.

Dalam penelitian ini, fitur yang digunakan untuk klasifikasi dibatasi pada fitur visual saja, yaitu fitur warna yang diekstraksi menggunakan

metode *Hue Saturation Value* (HSV), serta fitur tekstur yang diekstraksi dengan metode *Gray Level Co-occurrence Matrix* (GLCM). Fitur lain seperti bentuk, ukuran, atau informasi spasial tidak digunakan.

Metode klasifikasi yang digunakan juga dibatasi *pada K-Nearest Neighbor* (KNN) karena metode ini sederhana dan efektif dalam klasifikasi berbasis fitur. Selain itu, untuk mengurangi dimenasi data tanpa mengurangi informasi penting, digunakan metode *Principal Component Analysis* (PCA) sebagai satu-satunya teknik reduksi dimensi.

Seluruh proses dalam penelitian ini dilakukan menggunakan pengolahan citra, ekstraksi fitur, dan klasifikasi dilakukan menggunakan perangkat lunak MATLAB, yang dipilih karena kemampuannya dalam mengelola citra digital serta mendukung berbagai metode pengolahan dan analisis data secara efisien.

BAB II

LANDASAN TEORITIK

Dalam penelitian ini, digunakan beberapa konsep dan metode sebagai dasar dalam proses klasifikasi citra tanah. Pemahaman terhadap teori-teori tersebut penting agar proses analisis berjalan secara tepat dan menghasilkan output yang akurat. Klasifikasi citra tanah dilakukan untuk mengetahui jenis tanah berdasarkan tampilan gambar yang terlihat, sehingga diperlukan metode yang bisa mengambail ciri-ciri khusus dari gambar tersebut.

Empat metode yang digunakan dalam penelitian ini, yaitu *Hue Saturation Value* (HSV), *Gray Level Co-occurance Matrix* (GLCM), *Principal Component Analysis* (PCA) dan *K-Nearest Neighbor* (KNN). Setiap tahapan dijelaskan secara sistematis, mulai dari akuisisi data hingga klasifikasi, dengan menyesuaikan teknik perhitungan yang diterapkan dalam kode implementasi.

2.1 Citra Digital

Citra digital adalah gambar yang direpresentasikan dalam bentuk digital, sehingga bisa diproses oleh komputer. Citra tersusun dari elemen-elemen titik yang disebut piksel, yang membentuk baris dan kolom. Setiap piksel memiliki angka yang menunjukkan tingkat kecerahan atau warna pada titik tertentu dalam gambar.

Citra digital bisa diperoleh dari berbagai sumber, seperti kamera digital, scanner, atau simulasi komputer. Tujuan utama dari pengolahan citra digital adalah untuk memproses dan mengubah gambar agar kualitasnya lebih baik, mengambil informasi penting dari gambar, dan membuatnya lebih mudah dipahami atau digunakan dalam berbagai aplikasi [6].

2.2 MATLAB

MATLAB (*Matrix Laboratory*) merupakan sebuah perangkat lunak dan bahasa pemograman tingkat tinggi yang dirancang khusus untuk komputasi numerik, visualisasi data, dan pemograman. MATLAB dikembangkan dengan dasar pemikiran pemrosesan berbasis matriks, sehingga sangat efisien dalam menangani berbagai perhitungan matematis dan teknik, termasuk dalam bidang pengolahan citra digital (*image processing*) [7].

Dalam konteks pengolahan citra, MATLAB menyediakan berbagai *toolbox* dan fungsi yang sangat lengkap dan mudah digunakan. Fitur-fitur ini memungkinkan pengguna untuk melakukan bebagai operasi seperti penyesuaian tingkat kecerahan (*brightness*), kontras, filter spasial, segmentasi citra, transformasi geometris, hingga ekstraksi fitur citra. Dengan kemampuannya yang kuat dan antarmuka yang terintegrasi, MATLAB banyak digunakan dalam penelitian, pengembangan sistem pengenalan pola, klasifikasi citra,

serta aplikasi-aplikasi berbasis kecerdasan buatan yang memerlukakn pemrosesan data visual [8].

2.3 RGB

Ruang warna RGB adalah sistem warna yang terdiri dari tiga komponen utama, yaitu *Red* (R), *Green* (G), dan *Blue* (B). setiap warna dalam sistem ini dibentuk dari campuran ketiga warna dasar. Nilai masing-masing warna berada dalam rentang 0 hingga 255, dimana nilai yang lebih tinggi menunjukkan intensitas warna yang lebih terang [9].

RGB disebut sebagai warna primer aditif, karena berbagai warna lain dapat dihasilkan dengan mencampurkan ketiga warna dasar ini dalam proporsi tertentu. Sebagai contoh, Warna putih terbentuk ketika ketiga komponen R, G, dan B memiliki nilai maksimum (255, 255, 255) [9].

Ruang warna RGB merupakan sistem yang paling umum digunakan dalam perangkat digital, seperti layar monitor, televisi, kamera, serta dalam bidang pengolahan citra digital. Sistem ini menjadi standar universal karena mampu merepresentasikan warna dalam bentuk angka yang dapat dengan mudah diproses oleh komputer.

2.4 *Hue Staturation Value*

HSV (*Hue, Staturation, Value*) adalah model warna yang menggambarkan warna berdasarkan tiga komponen utama, yaitu *hue* (nuansa warna), *saturation* (kemurnian warna), dan *value* (tingkat kecerahan). Model ini dianggap lebih intutif dan sesuai dengan cara manusia memahami warna, dibandingkan dengan model warna RGB yang umum digunakan dalam representasi digital [8].

Berikut ini merupakan langkah-langkah atau metode yang dapat digunakan untuk mengonversi atau memperoleh nilai HSV dari data warna yang semula direpresentasikan dalam format RGB, sehingga warna dapat dianalisis dengan cara yang lebih intuitif dan mendekati presepsi manusia [10].

$$V = \max(r, g, b)$$

$$S = \left\{ \begin{matrix} 0, & jika\, Value = 0 \\ \left(\frac{\max - min}{max}\right), & jika\ lainnya \end{matrix} \right\}$$

$$H = \left\{ \begin{matrix} saturation = 0, hue = 0 \\ 60°\ x\ \left(\frac{G - B}{max - min} mod\ 6\right) \max = R \\ 60°\ x\ \left(\frac{B - R}{max - min} + 2\right), \max = G \\ 60°\ x\ \left(\frac{R - G}{max - min} + 4\right), \max = B \end{matrix} \right\}$$

Keterangan:

- S : nillai *saturation*

- V : nillai *value*
- H : nillai *hue*
- Max : nillai maximum dari normalisasi RGB
- Min : nillai minimum dari normalisasi RGB
- R : nillai kanal warna *red* (merah)
- G : nillai kanal warna *green* (hijau)
- B : nillai kanal warna *blue* (biru)

2.4.1 *Hue* (Nuansa Warna)

Hue menunjukkan jenis warna dasar seperti merah, hijau, biru dan sebagainya. Dalam model HSV, *hue* direpresentasikan dalam bentuk lingkaran warna, di mana setiap sudut pada lingkaran mewakili jenis warna berbeda. Nilai *hue* biasanya berada dalam rentang 0 hingga 360 derajat, dengan setiap rentang tertentu mewakili warna spesifik.

2.4.2 *Saturation* (Kemurnian Warna)

Saturation menunjukkan seberapa murni atau jenuh suatu warna. *Saturation* yang tinggi akan menghasilkan warna yang cerah dan pekat, sedangkan *saturation* yang rendah cenderung menghasilkan warna yang pucat atau cenderung ke abu-abu. Nilai *saturation* berada pada rentang 0 hingga 1, di mana 0 berarti warna sepenuhnya abu-abu (tidak jenuh), dan 1 berarti warna sangat jenuh.

2.4.3 *Value* (Tingkah Kecerahan)

Value menunjukkan seberapa terang atau gelap suatu warna. Nilai *value* yang tinggi akan menghasilkan warna yang terang, sedangkan nilai *value* yang rendah menghasilkan warna yang gelap atau redup. Sama seperti *saturation*, nilai *value* juga berada pada rentang 0 hingga 1.

2.4.4 Rumus HSV

Dari ketiga channel warna pada model HSV, yaitu *Hue* (H), *Saturation* (S), dan *Value* (V), dilakukan proses ekstraksi terhadap fitur-fitur statistik dasar yang bertujuan untuk menangkap karakteristik warna utama, tingkat kejenuhan, serta tingkat kecerahan dari setiap gambar. Ekstraksi ini menghasilkan representasi numerik yang digunakan dalam proses klasifikasi citra, sehingga model dapat mengenali pola warna yang khas dari masing-masing jenis tanah. Berikut adalah fitur statistik dasar yang dihitung:

2.4.4.1 Rata-rata (*mean*)

Rumus perhitungan rata-rata dari channel pada model warna HSV, digunakan untuk menghitung nilai tengah dari distribusi intensitas piksel pada masing-masing channel tersebut, dan dapat dituliskan sebagai berikut:

$$\boldsymbol{\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i}$$

Keterangan:

- μ_x = Nilai rata-rata dari data x
- N = Jumlah total data
- x_i **=** Nilai piksel ke-i pada channel tertentu (H, S, atau V)

2.4.4.2 Simpangan Baku (Standard Deviation)

Selain nilai rata-rata, fitur statistik lain yang penting untuk dianalisis dari setiap channel HSV adalah simpangan buku (standard deviation). Simpangan baku digunakan untuk mengukur tingkat penyebaran atau variasi nilai piksel terhadap nilai rata-ratanya dalam sebuah citra. Semakin besar nilai simpangan baku, maka semakin besar variasi warna yang terdapat pada channel tersebut. Rumus perhitungan simpangan baku ditulis sebagai berikut:

$$\boldsymbol{\sigma_x = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)^2}}$$

Keterangan:

- σ_x = Standar deviasi (menyatakan seberapa besar sebaran data terhadap rata-rata)

- N = Jumlah total data
- x_i = Nilai piksel ke-i pada channel tertentu (H, S, atau V)
- μ_x = Nilai rata-rata dari data x

Dalam konteks ini, $\boldsymbol{x}$ merupakan vektor nilai piksel dari salah satu channel HSV (*Hue, Saturation*, atau *Value*), dan ***N*** adalah jumlah total piksel dalam gambar. Melalui perhitungan ini, diperoleh total enam buah fitur dari model warna HSV, yaitu nilai rata-rata dan simpangan baku dari masing-masing channel H, S, dan V.

2.5 *Gray Level Co-Occurrence Matrix*

Gray Level Co-Occurrence Matrix (GLCM) merupakan metode yang umum digunakan dalam analisis tekstur atau ekstraksi ciri citra berbasis tekstur. Metode ini bekerja dengan menganalisis hubungan spasial antar piksel yang berdekat dalam gambar, lalu disusun dalam bentuk matriks berdasarkan arah dan jarak tertentu. Metode ini memiliki kelebihan, seperti proses yang cepat, sederhana, mudah dipelajari, dan tetap efektif meskipun digunakan pada jumlah data yang besar [11].

Sebelum menghitung GLCM dilakukan, citra terlebih dahulu dikonversi ke dalam bentuk *grayscale*. Setelah itu, GLCM dihitung berdasarkan empat arah, yaitu 0° (horizontal), 45° (diagonal kanan

atas), 90° (vertikal), dan 135° (diagonal kiri atas), dengan jarak antar piksel sebesar 1 piksel. Dari GLCM ini, dapat diperoleh informasi tekstur seperti tingkat kehalusan, kekerasan, atau keteraturan pola, yang kemudian digunakan sebagai fitur dalam proses klasifikasi.

Metode GLCM yang digunakan dalam penelitian ini melibatkan empat fitur utama yang digunakan untuk menggambarkan tektsur suatu citra, yaitu kontras, homogenitas, *energy*, dan korelasi [12].

2.5.1 Kontras (Contrast)

Kontras adalah fitur yang menggambarkan tingkat perbedaan antara piksel terang dan gelap dalam sebuah citra. Nilai kontras akan tinggi jika perbedaan intentitas antara piksel-piksel berdekatan juga tinggi. Fitur ini membantu mengidentifikasi batas-batas atau tekstur yang tajam pada gambar. Nilai kontras dihitung menggunakan rumus berikut:

$$\boldsymbol{Kontras} = \sum_{\boldsymbol{i,j}} (\boldsymbol{i} - \boldsymbol{j})^2 \,.\, \boldsymbol{P}(\boldsymbol{i}, \boldsymbol{j})$$

Keterangan:

- i, j = Nilai tingkat keabuan (*grayscale level*)
- $P(i, j)$ = Nilai probabilitas pasangan piksel (i, j) dalam matriks GLCM
- Nilai kontras berkisar dari 0 (tidak ada perbedaan) hingga nilai maksimum tergantung pada rentang tingkat keabuan citra

2.5.2 Homogenitas (*Homogenity*)

Homogenitas mengukur seberapa merata atau serangam intensitas piksel dalam citra. Nilai homogenitas akan tinggi jika intensitas piksel antara pasangan piksel saling mendekati atau tidak banyak perbedaan, yang mencerminkan tekstur yang halus dan seragam. Rumus perhitungan homogenitas sebagai berikut:

$$\textbf{Homogenitas} = \sum_{i,j} \frac{P(i,j)}{1 + |i - j|}$$

Keterangan:

- i, j = Nilai tingkat keabuan (grayscale level)
- $P(i,j)$ = Nilai probabilitas pasangan piksel (i,j) dalam matriks GLCM
- Semakin kecil perbedaan antara i dan j, maka semakin besar kontribusinya terhadap nilai homogenitas
- Nilai maksimal dicapai saat seluruh piksel memiliki nilai intensitas yang sama

2.5.3 *Energy* (ASM)

Energy atau dikenal juga dengan *Angular Second Moment* adalah ukuran keteraturan dalam distribusi tekstur. Nilai *energy* tinggi menunjukkan bahwa gambar memiliki pola tekstur yang konsisten dan tidak acak. Rumus perhitungan *energy* sebagai berikut:

$$\boldsymbol{Enery} = \sum_{i,j} \boldsymbol{P(i,j)^2}$$

Keterangan:

- i, j = Nilai tingkat keabuan (*grayscale level*)
- $P(i, j)$= Nilai probabilitas pasangan piksel (i, j) dalam matriks GLCM
- Semakin tinggi nilai energy, semakin seragam atau teraturnya citra
- Cocok untuk mendeteksi gambar dengan struktur atau pola yang berulang

2.5.4 Korelasi (*Correlation*)

Korelasi mengukur hubungan adalah linier antara pasangan nilai piksel dalam citra. Nilai korelasi menunjukkan apakah piksel-piksel memiliki hubungan yang konsisten dalam arah tertentu. Nilai ikorelasi berada pada rentang antara -1 hingga 1. Rumus perhitungan korelasi sebagai berikut:

$$\boldsymbol{Korelasi} = \sum_{i,j} \frac{\boldsymbol{(i - \mu_i)(j - \mu_j)P(i,j)}}{\boldsymbol{\sigma_i \sigma_j}}$$

Keterangan:

- i, j = Nilai tingkat keabuan (*grayscale level*)
- $P(i, j)$ = Nilai probabilitas pasangan piksel (i, j) dalam matriks GLCM

- μ_i, μ_j = Rata-rata tingkat keabuan pada baris dan kolom
- σ_i, σ_j = Simpangan baku tingkat keabuan pada baris dan kolom
- Nilai korelasi mendekati 1 menunjukkan hubungan linier yang kuat antara piksel

Setiap fitur tektur, seperti kontras, homogenitas, *energy*, dan korelasi, dihitung berdasarkan empat arah GLCM, yaitu 0°, 45°, 90°, dan 135°. Setelah itu, hasil dari keempat arah tersebut dirata-rata agar didapat satu nilai untuk masing-masing fitur. Jadi, total ada empat fitur tekstur yang digunakan untuk membantu proses klasifikasi gambar.

2.6 *Principal Component Analysis*

Principal Component Analysis (PCA) adalah metode matematis yang digunakan untuk melakukan reduksi dimensi data, dengan cara mengubah data berdimensi tinggi menjadi data berdimensi lebih rendah tanpa kehilangan informasi penting secara signifikan. PCA berasal dari bidang statistik, tetapi penerapannya juga sangat luas dalam pengolahan data, termasuk dalam pengolahan citra digital, terutama pada tahap ekstraksi ciri [13].

Dalam penelitian ini, fitur gabungan yang dihasilkan dari metode HSV dan GLCM berubah menjadi 10 dimensi. Terdiri atas 6 fitur dari model warna HSV (rata-rata dan standar devisiasi dari masing-masing saluran warna *Hue*, *Saturation*, dan *Value*), serta 4 fitur dari GLCM

(kontras, homogenitas, *energy*, dan korelasi). Untuk menyederhanakan data dan mengurangi kompleksitas, serta menghindari overfitting saat pelatihan model klasifikasi, dilakukan proses reduksi dimensi menggunakan PCA. Proses ini dilakukan melalui beberapa tahapan berikut:

2.6.1 Normalisasi Data

Data awal dikurangi dengan nilai rata-rata dari setiap fitur agar setiap fitur memiliki distribusi yang terpusat di sekitar nol. Langkah ini penting dilakukan untuk menghilangkan bias akibat perbedaan skala antara fitur, serta memastikan bahwa hasil perhitungan matriks kovarians dan proyeksi ke dimensi baru menjadi lebih akurat dan representatif.

$$\boldsymbol{Z} = \boldsymbol{X} - \overline{\boldsymbol{X}}$$

Keterangan:

- X = Data asli
- $\overline{X}$ = Nilai rata-rata fitur
- Z = Hasil normalisasi

2.6.2 Kalkulasi Matriks Kovarians

Setelah dilakukan normalisasi data, langkah selanjutnya dalam proses *Principal Componen Analysis* (PCA) adalah menghitung matriks kovarians. Matriks ini digunakan untuk

menggambarkan hubungan atau keterkaitan antara fitur (variabel) dalam data. Dengan kata lain, matriks kovarians menunjukkan seberapa besar dua fitur berubah bersama-sama. Jika dua fitur cenderung meningkat atau menurun bersamaan, maka kovariansnya bernilai positif. Sebaliknya, jika satu meningkat sementara yang lain menurun, maka kovariansnya bernilai negatif. Rumus untuk menghitung matriks kovarians adalah sebagai berikut:

$$\boldsymbol{C} = \frac{\mathbf{1}}{\boldsymbol{n} - \mathbf{1}} \boldsymbol{Z}^{\boldsymbol{T}} \boldsymbol{Z}$$

Keterangan:

- C = Matriks kovarians
- n = Jumlah sampel data
- Z = Hasil normalisasi
- Z^T = Transpose dari data normalisasi

Melalui perhitungan ini, diperoleh informasi penting tentang struktur data, yang kemudian digunakan untuk menentukan arah (komponen utama) terbaik dalam mereduksi dimensi tanpa kehilangan informasi yang berarti.

2.6.3 Menghitung Nilai dan Vektor Eigen

Langkah selanjutnya dalam proses *Principal Component Analysis* (PCA) adalah menghitung nilai eigen (*eigenvalue*) dan

vektor eigen (*eigenvector*) dari matriks kovarians yang telah diperoleh sebelumnya. Nilai eigen berfungsi untuk menunjukkan seberapa besar variasi atau informasi dari data yang bisa dijelaskan oleh masing-masing komponen utama. Semakin besar nilai eigen, maka semakin besar pula informasi yang dibawa oleh komponen tersebut.

Sementara itu, vektor eigen adalah arah dari komponen utama, yang menentukan sumbu baru tempat akan diproyeksikan. Vektor ini mewakili garis arah utama distribusi data, dan akan digunakan sebagai dasar dalam membentuk dimensi baru yang lebih ringkas namun tetap informatif.

2.6.4 Proyeksi ke Dimensi Baru

Setelah diperoleh semua nilai vektor eigen, tahap berikutnya adalah memilih sejumlah vektor eigen dengan nilai eigen terbesar. Vektor-vektor ini dipilih kakrena membawa informasi paling penting dari keseluruhan data.

Vektor-vektor tersebut kemudian digunakan untuk membentuk ruang fitur baru (ruang dimensi yang telah direduksi). Seluruh data kemudian diproyeksikan ke dalam ruang ini, sehingga menghasilkan representasi baru dari data dengan dimensi yang lebih rendah. Meskipun jumlah dimensinya berkurang, data hasil proyeksi tetap mempertahankan

karakteristik utama atau informasi paling relavan dari data awal, sehingga proses analisis atau klasifikasi tetap bisa dilakukan secara efektif namun lebih efisien.

2.7 *K-Nearest Neighbor*

K-Nearest Neighbor (KNN) merupakan salah satu metode klasifikasi yang sederhana, namun efektif. Metode ini bekerja dengan cara membandingkan data uji dengan data latih berdasarkan jarak dekat. Dalam penelitian ini, proses klasifikasi dilakukan setelah fitur citra direduksi menggunakan metode *Principal Component Analysis* (PCA), sehingga data yang digunakan lebih ringkas namun tetap menyimpan informasi penting.

Fitur yang digunakan dalam klasifikasi KNN adalah hasil dari ekstraksi ciri menggunakan metode HSV dan GLCM, yang kemudian di reduksi dimensinya menggunakan *eigenvector* melalui PCA [13]. Berikut merupakan langkah-langkah klasifikasi dengan KNN:

2.7.1 Menghitung Jarak *Euclidean*

Langkah pertama dalam proses klasifikasi menggunakan metode KNN adalah menghitung jarak antara citra uji dan seluruh citra latih. Tujuannya adalah untuk mengetahui seberapa mirip atau dekat suatu citra uji terhadap data latih berdasarkan nilai-nilai fitur yang dimiliki.

Untuk mengukur kedekatan tersebut, digunakan rumus jarak *euclidean*, yaitu salah satu metode pengukuran jarak yang paling umum dan sederhana. Jarak euclidean menghitung akar kuadrat dari jumlah selisih kuadrat antara nilai-nilai fitur citra uji dengan nilai-nilai fitur citra latih pada setiap dimensi fitur. Rumus jarak *euclidean* sebagai berikut:

$$\boldsymbol{d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}}$$

Keterangan:

- $d(x,y)$ = Jarak *Euclidean* antara citra uji x dan citra latih y
- x_i = Nilai fitur ke-i dari citra uji
- y_i = Nilai fitur ke-i dari citra tanah
- n = Jumlah total fitur yang digunakan

Semakin kecil nilai jarak *euclidean* yang diperoleh, maka semakin dekat atau mirip citra uji dengan citra latih. Jarak ini kemudian menjadi dasar untuk memilih tetangga terdekat dalam menentukan hasil klasifikasi.

2.7.2 Menentukan k Tetangga Tedekat

Setelah jarak antara citra uji dan seluruh citra latih dihitung menggunakan rumus jarak *euclidean,* langkah selanjutnya adalah menentukan jumlah k tetangga terdekat dari citra uji.

Tetangga terdekat ini adalah citra-citra latih yang memiliki nilai jarak paling kecil terhadap citra uji, sehingga dianggap paling mirip berdasarkan fitur yang telah diekstraksi.

Sebagai contoh, jika digunakan nilai k = 5, maka lima citra latih dengan jarak terdekat ke citra uji akan dipilih sebagai acuan untuk menentukan kelas. Pemilihan nilai k yang sesuai sangat penting karena dapat mempengaruhi hasil klasifikasi. Nilai k yang terlalu kecil dapat menyebabkan hasil menjadi sensitif terhadap data tidak normal (*noise*), sedangkan nilai k yang terlalu besar dapat mengurangi kemampuan model dalam membedakan antara kelas.

2.7.3 Menentukan Kelas Mayoritas

Setelah k tetangga terdekat diperoleh, langkah berikutnya adalah menentukan yang paling sering muncul dari kelompok tetangga tersebut. Proses ini dikenal sebagai *majority voting* atau pemungutan suara mayoritas. Citra uji akan diklasifikasikan ke dalam kelas yang paling banyak muncul di antara k tetangga tadi. Sebagai ilustrasi, jika dari 5 tetangga terdekat terdapat 3 citra yang termasuk dalam kelas "*black soil*" dan 2 citra lainnya termasuk dalam kelas "*yellow soil*", maka citra uji akan diklasifikasikan ke dalam kelas "*black soil*". Langkah ini merupakan tahap akhir dari proses klasifikasi

menggunakan metode KNN, yang bergantung pada kemiripan fitur dan jumlah kemunculan kelas dari citra-citra latih terdekat.

Karakteristik utama dari metode *K-Nearest Neighbor* (KNN) adalah bersifat non-parametrik, yang berarti tidak membuat asumsi khusus terhadap bentuk distribusi data. Hasl ini menjadikan KNN fleksibel dalam menghadapi berbagai jenis data. Pemilihan nilai k sangat berpengaruh terhadap kinerja model, jika nilai k terlalu kecil, model bisa menjadi terlalu sesnsitif terhadap data noise atau pencilan, sedangkan nilai k yang terlalu besar dapat menyebabkan penurunan akurasi karena model terlalu umum dalam menentukan kelas. Selain itu, KNN sangat cocok digunakan pada dataset yang berukuran kecil hingga sedang, serta memiliki keunggulan dalam hal kemudahan implementasi dan pemahaman.

BAB III

METODE EKSPERIMEN

Bab ini menjelaskan secara rinci tahapan-tahapan yang dilakukan dalam proses eksperimen penelitian, mulai dari proses input data, tahap pra-pemrosesan, ekstraksi fitur warna menggunakan metode HSV, ekstraksi fitur tekstur dengan metode GLCM, reduksi dimensi menggunakan PCA, Klasifikasi menggunakan algoritma KNN, hingga penyusunan diagram alir yang menggambarkan keseluruhan alur poses penelitian.

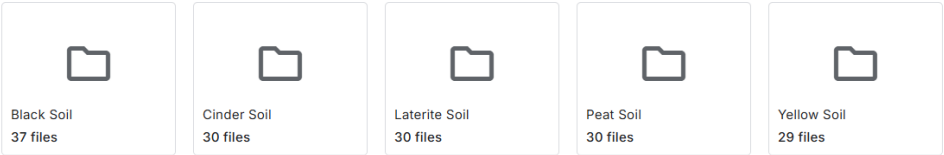
3.1 Pengumpulan Data

Data yang digunakan dalam penelitian ini berasal dari dataset citra tanah yang diperoleh dari situs *Kaggle*, sebuah platform berbagi data yang sering digunakan dalam penelitian berbasis mesin dan pengolahan citra digital. Dataset ini berjudul “Soil Types” dan disusun oleh pengguna dengan nama akun PRASANSHA SATPATHY, yang telah diunggah sekitar empat tahun yang lalu. Dataset tersebut dapat diakses secara publik melalui tautan berikut: https://www.kaggle.com/datasets/prasanshasatpathy/soil-types.

Dataset ini terdiri atas 156 gambar citra tanah yang dikelompokkan ke dalam lima kategori berdasarkan jenis tanah, yaitu *Black Soil*, *Cinder Soil*, *Laterite Soil*, *Peat Soil*, dan *Yellow Soil*. Dataset ini dipilih karena memiliki data yang cukup beragam dan

mencakup beberapa jenis tanah yang umum ditemukan. Selain itu, format dan struktur file pada dataset ini mendukung untuk digunakan dalam proses pelatihan model klasifiasi.

Gambar-gambar pada dataset disimpan dalam format .jpg dengan kualitas yang cukup baik, namun ukuran piksel dari setiap gambar masih bervariasi dan tidak seragam, sehingga diperlukan penyesuaian ukuran sebelum proses pelatihan model. Proses ini penting agar semua gambar memiliki resolusi yang konsisten dan sesuai dengan input layer pada model yang digunakan. Tampilan dataset jenis tanah dapat dilihat pada Gambar 3.1.1.



Gambar 3.1. 1 Dataset citra tanah sebelum sotir

3.1.1 Black Soil

Tanah hitam atau *black soil* memiliki warna gelap yang khas, kaya akan bahan organik dan mineral. Jenis tanah ini memiliki daya serap air yang tinggi dan sering ditemukan di daerah tropis atau subtropis. Tanah ini cocok untuk pertanian, terutama untuk tanaman kapas dan jagung. Karakteristik *black soil* dapat dilihat pada Tabel 3.1.1.1 dan gambar dataset *black soil* dapat dilihat pada Gambar 3.1.1.1.

Tabel 3.1.1. 1 Karakteristik Black Soil

Sifat	Keterangan
Warna	Hitam pekat
Tekstur	Halus, sedikit lengket saat basah
Kandungan organik	Tinggi, terutama humus
Struktur	Cenderung berpori dan baik untuk penyerapan air

Gambar 3.1.1. 1 Tampilan beberapa dataset black soil sebelum pre-processing

3.1.2 Cinder Soil

Tanah *cinder soil* berasal dari material vulkanik seperti abu dan batuan lava yang telah hancur. Tanah ini memiliki warna keabu-abuan cokelat tua, dan teksturnya ringan serta berpori. Tanah ini sering ditemukan di daerah bekas letusan gunung berapi. *Cinder soil* tidak banyak ditemukan di wiayah Indonesia. Karakteristik *cinder soil* dapat dilihat pada Tabel

3.1.2.1, dan gambar dataset *cinder soil* ditampilkan pada Gambar 3.1.2.1.

Tabel 3.1.2. 1 Karakteristik Cinder Soil

Sifat	Keterangan
Warna	Abu-abu hingga cokelat tua
Tekstur	Ringan, berpori
Kandungan organik	Rendah
Struktur	Longgar, sangat mudah mengalirkan air

Gambar 3.1.2. 1 Tampilan beberapa dataset cinder soil sebelum pre-processing

3.1.3 Laterite Soil

Tanah laterit atau *laterite soil* terbentuk melaluii proses pelapukan batuan dalam kondisi iklim tropis yang panas dan lembap. Tanah ini kaya akan kandung zat besi dan aluminium,

yang memberikan warna merah hingga cokelat kemerahan. Tanah ini umumnya ditemukan di daerah dataran tinggi berbatu dan sering digunakan untuk pembangunan atau ditanami tanaman keras. Karakteristik *laterite soil* dapat dilihat pada Tabel 3.1.3.1, dan gambar dataset *laterite soil* ditampilkan pada Gambar 3.1.3.1.

Tabel 3.1.3. 1 Karakteristik Laterite Soil

Sifat	Keterangan
Warna	Merah hingga merah kecokelatan
Tekstur	Kasar dan keras saat kering
Kandungan organik	Tinggi, terutama oksida besi dan aluminium
Struktur	Padat, air cepat meresap, tidak terlalu subur

Gambar 3.1.3. 1 Tampilan beberapa dataset laterite soil sebelum pre-processing

3.1.4 Peat Soil

Tanah gambut atau *peat soil* terbentuk dari akumulasi bahan organik yang belum terdekomposisi sepenuhnya, biasanya berada di daerah berawa. Tanah ini sangat kayak akan bahan organik, memiliki kandungan air tinggi, dan sangat asam. *Peat soil* sering digunakan untuk pertanian, tetapi memerlukan pengelolaan air yang baik. Karakteristik *peat soil* dapat dilihat pada Tabel 3.1.4.1, sedangkan tampilan dataset *peat soil* dapat dilihat pada Gambar 3.1.4.1.

Tabel 3.1.4. 1 Karakteristik Peat Soil

Sifat	Keterangan
Warna	Gelap hingga hitam
Tekstur	Lembek dan basah
Kandungan organik	Sangat tinggi
Struktur	Ringan dan bisa cepat terbakar saat kering

Gambar 3.1.4. 1 Tampilan beberapa dataset peat soil sebelum pre-processing

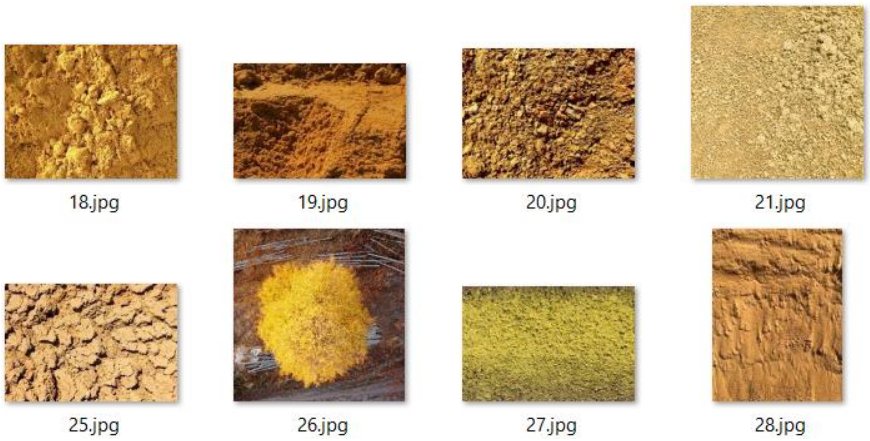
3.1.5 Yellow Soil

Tanah kuning atau *yellow soil* berwarna kuning cerah hingga coklat muda karena kandungan oksida besi yang lebih rendah. Tanah ini baisanya ditemukan didaerah berbukit dan memiliki kemampuan menyerap air yang baik, meskipun tingkat kesuburannya lebih rendah dibanding *black soil* atau *peat soil*. Karakteristik *yellow soil* dapat dilihat pada Tabel 3.1.5.1, dan contoh citra dataset *yellow soil* dapat dilihat pada Gambar 3.1.5.1.

Tabel 3.1.5. 1 Karakteristik Yellow Soil

Sifat	Keterangan
Warna	Kuning terang hingga cokelat muda
Tekstur	Sedikit kasar

Kandungan organik	Rendah, terutama pada unsur hara
Struktur	Air mengalir dengan baik, subur jika dipupuk



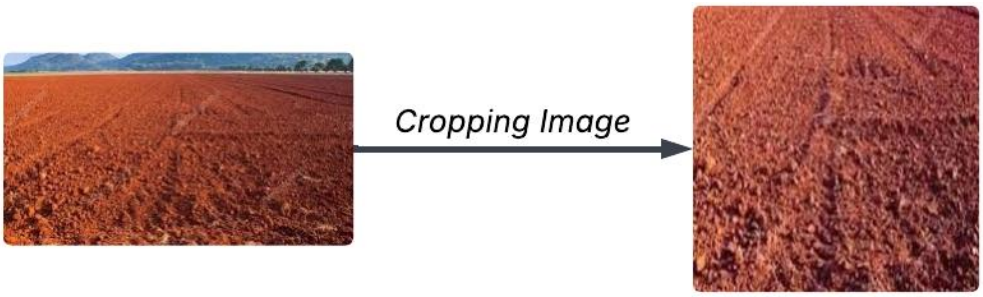
Gambar 3.1.5. 1 Tampilan beberapa dataset yellow soil sebelum pre-processing

3.2 Pra-Pemrosesan

Pada tahap awal pra-pemrosesan, dilakukan pemilihan jenis tanah yang akan digunakan dalam penelitian. Dari lima ketegori tanah yang tersedia dalam dataset, hanya empat jenis tanah yang dipilih, yaitu *Black Soil*, *Laterite Soil*, *Peat Soil*, dan *Yellow Soil*. Sementara itu, ketegori *Cinder Soil* tidak digunakan karena jenis tanah tersebut jarang ditemukan di wilayah Indonesia, sehingga dianggap tidak relavan untuk dijadikan objek penelitian. Setelah penghapusan ketegori tersebut, jumlah gambar berkurang menjadi 126 gambar.

Langkah selanjutnya adalah proses kurasi data untuk memastikan kualitas gambar sesuai dengan kebutuhan penelitian. Beberapa gambar dihapus secara manual karena memiliki kualitas rendah dan tidak merepresentasikan objek tanah dengan baik. Setelah proses penyaringan, tersisa 120 gambar yang layak digunakan untuk pelatihan dan pengujian.

Untuk meningkatkan fokus area tanah, sebagian gambar dilakukan proses *cropping* untuk menghilangkan bagian yang tidak relavan, seperti latar belakang atau objek lain diluar tanah. Selain itu, seluruh gambar diubah ukurannya secara manual menjadi 224 x 224 piksel guna menyeragamkan dimensi input sebelum dimasukkan ke dalam model klasifikasi. Contoh gambar hasil proses *cropping* dapat dilihat pada Gambar 3.2.1. Contoh gambar hasil perubahan ukuran dapat dilihat pada Gambar 3.2.2, Gambar 3.2.3, Gambar 3.2.4, dan Gambar 3.2.5.

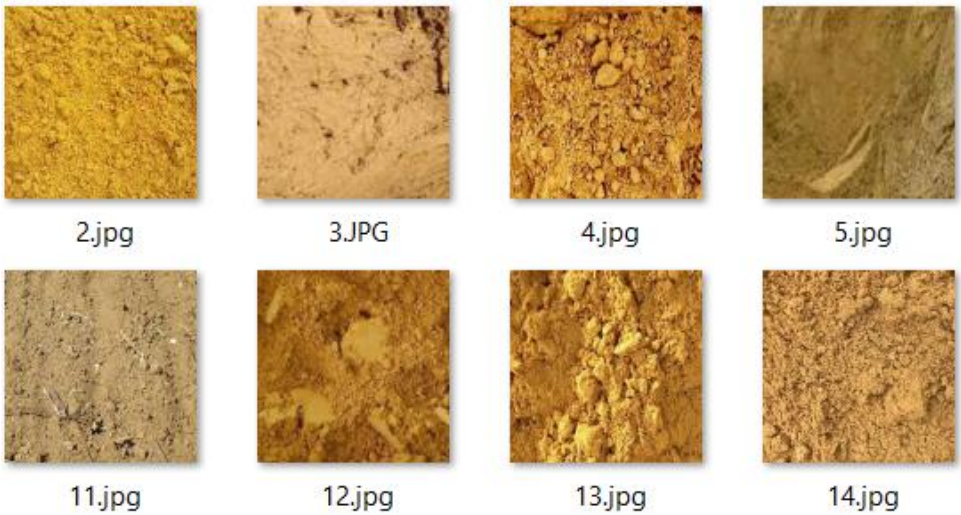


Gambar 3.2. 1 Proses Cropping salah satu Gambar

Gambar 3.2. 2 Hasil Resize 224 x 224 piksel black soil

Gambar 3.2. 3 Hasil Resize 224 x 224 piksel laterite soil

Gambar 3.2. 4 Hasil Resize 224 x 224 piksel peat soil



Gambar 3.2. 5 Hasil Resize 224 x 224 piksel yellow soil

Karena jumlah gambar yang tersedia masih terbatas, dilakukan proses augmentasi data untuk menambah jumlah dan variasi citra dalam dataset. Teknik augmentasi yang digunakan meliputi rotasi, pembalikan horizontal (flip horizontal), dan pembalikan vertikal (flip vertikal). Proses ini bertujuan untuk meningkatkan keberagamaan data agar model dapat belajar dari berbagai sudut pandang citra.

Setelah proses augmentasi selesai, jumlah total gambar meningkat meningkat menjadi 429 gambar. Dataset yang telah diperluas ini kemudian dibagi menjadi dua bagian, yaitu data pelatihan dan data pengujian. Persentase pembagian data dapat dilihat pada Tabel 3.2.1.

Table 3.2. 1 Pembagian Data Pelatihan dan Data Pengujian

Nama Data	Persentasi Pembagian Data
Data Pelatihan (Training Set)	70%
Data Pengujian (Testing Set)	30%

Setelah dataset citra tanah diperoleh dan disiapkan, langkah selanjutnya adalah membaca gambar-gambar dari folder utama beserta seluruh subfolder yang ada. Untuk proses ini digunakan fungsi **imageDatastore** dalam MATLAB, yang berfungsi untuk memuat data citra secara otomatis berdasarkan struktur folder yang tersedia. Potongan kode yang digunakan untuk membaca dataset dapat dilihat pada pada potongan kode berikut:

```
% Step 1: Load image dataset
imds = imageDatastore('soil-types', ...
% Membuat sebuah datastore gambar dari folder 'soil-types'
    'IncludeSubfolders', true, ...
% Menyertakan semua subfolder di dalam 'soil-types'
    'LabelSource', 'foldernames');
% Secara otomatis memberi label pada gambar berdasarkan nama foldernya
```

Setelah proses pembacaan file gambar dilakukan menggunakan **imageDatastore**, dilakukan perhitungan jumlah gambar serta inisialisasi array kosong untuk menyimpan hasil ekstraksi fitur dan label. Potongan kode ini penting sebagai fondasi sebelum dilakukan proses konversi citra dan ekstraksi fitur. Potongan kode perhitungan jumlah gambar dapat di lihat pada potongan kode berikut:

```
% Step 2: Ekstraksi fitur warna (HSV) dan tekstur (GLCM)
numImages = numel(imds.Files); %numImages memuat jumlah
gambar di dataset
features = []; %array kosong untuk menyimpan fitur
labels = []; %array kosong untuk label jenis tanah
```

potongan kode berikut bertujuan untuk menghitung total jumlah citra yang akan diproses dalam dataset menggunakan variabel **numImages** yang menjalankan fungsi *built-in* dari matlab sendiri yaitu numel() yang berfungsi untuk menghitung jumlah total elemen dalam sebuah array [14]. Selain itu, disiapkan dua array kosong, yaitu **features** yang akan digunakan untuk menampung hasil ekstraksi fitur dari setiap gambar, baik fitur warna maupun tekstur, serta **labels** yang digunakan untuk menyimpan informasi label atau kategori jenis tanah dari masing-masing gambar. Inisialisasi ini merupakan langkah penting sebelum data diproses lebih lanjut dalam tahap ekstraksi fitur dan klasifikasi.

3.3 Ekstraksi Warna HSV

Selanjutnya adalah membaca gambar menggunakan fungsi *built-in* dari Matlab yaitu **readImage()** untuk mengimport file gambar [14] yang berfungsi untuk Setelah gambar dibaca menggunakan fungsi **readImage**, tahap selanjutnya adalah melakukan konversi gambar ke dalam ruang warna HSV (Hue, Saturation, dan Value) menggunakan fungsi *built-in* **rgb2hsv()** yang berfungsi mengubah nilai merah, hijau, dan biru dari gambar RGB menjadi nilai rona, saturasi, dan nilai (HSV) dari gambar HSV [14] . Ruang warna HSV dipilih karena lebih mendekati persepsi warna manusia dibandingkan format RGB. Potongan kode untuk proses pembacaan gambar **readImage** ditampilkan pada potongan kode berikut:

```
img = readimage(imds, i); % membaca gambar ke i, dimasukkan ke img
```

Masing-masing channel pada gambar HSV yaitu Hue (H), Saturation (S), dan Value (V), kemudian dianalisis secara statistik. Dari setiap channel, dihitung nilai rata-rata (*mean*) dan standar deviasi (*standard deviation*). Dengan demikian, diperoleh total enam fitur warna dari satu gambar, yang terdiri dari: mean-H, std-H, mean-S, std-S, mean-V, dan std-V. Potongan kode konversi gambar, pengambilan channel HSV, serta perhitungan nilai rata-rata dan standar devisiasi

pada masing-masing channel ditampilkan pada potongan kode berikut:

```
hsvImg = rgb2hsv(img); %konversi gambar menjadi ruang warna HSV
% Kemudian mengambil nilai rata dan standar deviasi dari masing masing H, S, dan V
% Sehingga total terdapat 6 fitur
H = hsvImg(:,:,1); S = hsvImg(:,:,2); V = hsvImg(:,:,3); % ambil semua baris (:), ambil semua
kolom (:), ambil channel ke (1-2-3)
hsvFeatures = [mean(H(:)), std(H(:)), ...
            mean(S(:)), std(S(:)), ...
            mean(V(:)), std(V(:))];
```

3.4 Ekstraksi GLCM

Setelah fitur warna berhasil diekstraksi, tahap selanjutnya adalah mengekstraksi fitur tekstur dari citra yang sama. untuk proses ini, gambar terlebih dahulu dikonveri dari format RGB ke format *greyscale* menggunakkan fungsi *built-in* Matlab yaitu **rgb2gray()** yang memiki fungsi untuk mengekstraksi fitur tekstur GLCM [14], karena metode Gray Level Co-occurrence Matrix (GLCM) hanya dapat diterapkan pada gambar tingkat keabuan. Potongan kode konversi gambar ke *grayscale* ditampilkan pada potongan kode berikut:

```
grayImg = rgb2gray(img); %konversi gambar ke grayscale
```

GLCM digunakan untuk menganalisis tekstur citra berdasarkan hubungan intensitas piksel yang berdekatan dalam arah tertentu. Dalam penelitian ini, digunakan empat arah utama, yaitu 0°, 45°, 90°,

dan 135°, dengan parameter **Offset** tertentu. Potongan kode untuk pembuatan GLCM berikut ini:

```
glcm = graycomatrix(grayImg, 'Offset', [0 1; -1 1; -1 0; -1 -1]);
```

Berdasarkan matriks GLCM yang dihasilkan, dihitung empat properti tekstur utama: *Contrast*, *Correlation*, *Energy*, dan *Homogeneity*. Karena GLCM dihitung dalam empat arah, maka setiap properti menghasilkan empat nilai. Potongan kode perhitungan properti GLCM dapat dilihat berikut ini:

```
stats = graycoprops(glcm, {'Contrast', 'Correlation', 'Energy', 'Homogeneity'}); % Hitung
properti tekstur GLCM (Contrast, Correlation, Energy, Homogeneity) dari matriks glcm.

 % Ambil rata-rata dari 4 arah GLCM
contrast = mean(stats.Contrast);
% Rata-rata nilai Contrast dari semua arah.
correlation = mean(stats.Correlation);
% Rata-rata nilai Correlation dari semua arah.
energy = mean(stats.Energy);
% Rata-rata nilai Energy dari semua arah.
homogeneity = mean(stats.Homogeneity);
% Rata-rata nilai Homogeneity dari semua arah.
glcmFeatures = [contrast, correlation, energy, homogeneity];
% Gabungkan rata-rata properti GLCM menjadi satu array fitur.
```

Untuk mendapatkan representasi tekstur yang lebih umum dan stabil, dilakukan pengambilan nilai rata-rata dari masing-masing properti. Dengan demikian, diperoleh total empat fitur tekstur dari satu gambar, yaitu: *mean-contrast*, *mean-correlation*, *mean-energy*, dan *mean-homogeneity*. Keempat nilai tersebut kemudian dikombinasikan

dengan fitur warna untuk tahap selanjutnya yaitu PCA dan klasifikasi. Potongan akhir penggabungan fitur pada kode berikut:

```
% Gabungkan fitur
totalFeatures = [hsvFeatures, glcmFeatures]; % Menggabungkan fitur HSV dan GLCM
menjadi satu array fitur tunggal untuk gambar saat ini.
features = [features; totalFeatures]; % Menambahkan array 'totalFeatures' (fitur dari gambar
saat ini) sebagai baris baru ke array 'features' yang menyimpan semua fitur dari dataset.
labels = [labels; imds.Labels(i)]; % Menambahkan label dari gambar saat ini ke array 'labels'
yang menyimpan semua label.
```

3.5 Reduksi PCA

Setelah fitur warna dan fitur tekstur diekstraksi dari seluruh gambar dalam dataset, tahap selanjutnya adalah melakukan reduksi dimensi menggunakan metode Principal Component Analysis (PCA). Reduksi dimensi ini bertujuan untuk menyederhanakan representasi data dengan tetap mempertahankan informasi pentinng dari fitur aasli, sekaligus mengurangi kompleksitas komputasi pada tahap klasifikasi.

Sebelum diterapkan PCA, seluruh data fitur terlebih dahulu dilakukan proses standardisasi, yaitu dengan mengurangkan nilai rata-rata dan membaginya dengan standar deviasi masing-masing fitur. Langkah ini penting untuk memastikan bahwa seluruh fitur memiliki skala yang sebanding dan tidak mendominasi dalam penbentukan komponen utama. Potongan kode proses standardisasi dapat dilihat pada potongan kode berikut:

```
% Step 3: PCA (standarisasi fitur HSV + GLCM)
Z = features - mean(features);      % Kurangi mean tiap fitur
```

```
stddev = std(Z);            % Hitung standar deviasi tiap fitur
Zr = Z ./ stddev;           % Standardisasi (bagi dengan std dev)
```

Setelah proses standarisasi, dihitung matriks kovarians dari data fitur, kemudian dilakukan dekomposisi eigen terhadap matriks tersebut untuk memperoleh eigenvalue dan eigenvector. Eigenvalue digunakan untuk menentukan proporsi varians yang dijelaskan oleh masing-masing komponen utama. Potongan kode perhitungan matriks kovarians dan dekomposisi eigen ditampilkan pada potongan kode berikut:

```
% Matriks kovarians dan eigen decomposition
Zcov = cov(Zr); % Hitung matriks kovarians dari data terstandardisasi (Zr).
[V, D] = eig(Zcov); % Lakukan dekomposisi eigen untuk mendapatkan vektor eigen (V) dan
nilai eigen (D).
```

Nilai eigen dan vektor eigen tersebut kemudian diurutkan dari yang terbesar hingga terkecil untuk mengetahui komponen utama mana yang paling penting. Dalam penelitian ini, dipilih jumlah komponen utama yang mampu menjelaskan setidaknya 90% dari total varians data. Komponen utama yang terpilih kemudian digunakan untuk memproyeksikan data ke ruang dimensi yang lebih rendah. Pemotongan kode untuk poses pengurutan dan proyeksi data dapat dilihat pada potongan kode berikut:

```
% Urutkan eigenvalue dan eigenvector dari terbesar ke terkecil
p = size(features, 2); % Jumlah fitur.
```

```
m = 1; % Indeks untuk pengurutan.
for i = p:-1:1
   DA_urut(m,m) = D(i,i); % Urutkan eigenvalue menurun.
   VA_urut(:,m) = V(:,i); % Urutkan eigenvector sesuai eigenvalue.
   m = m + 1;
end

% Hitung proporsi varians dan pilih komponen utama yang total 90%
for i = 1:p
   eValue(i,1) = DA_urut(i,i); % Ambil eigenvalue terurut.
end

Total = sum(eValue); % Total varians.
PK = 0; % Kumulatif varians dijelaskan.
komponen = 0; % Jumlah komponen terpilih.
for j = 1:p
   proporsiValue(j,:) = eValue(j,1) / Total; % Proporsi varians per komponen.
   if PK < 0.9 % Jika kumulatif < 90%.
      PK = PK + proporsiValue(j,:); % Tambah kumulatif.
      komponen = komponen + 1; % Tambah komponen terpilih.
   end
end

% Proyeksi data ke ruang PCA
VA_Komponen = VA_urut(:,1:komponen); % Ambil eigenvector komponen terpilih.
W = Zr * VA_Komponen; % Proyeksikan data ke ruang PCA.
```

Untuk mendukung analisis visual, dilakukan visual 3D dari hasil proyeksi PCA menggunakan scatter plot tiga dimensi. Visualisasi ini menggambarkan distribusi masing-masing kelas tanah dalam ruang fitur hasil PCA dan memberikan gambaran awal megenai separabilitas antar kelas. Potongan kode visualisasi proyeksi PCA sebagai berikut:

```
%% 🔍 Tambahan Visualisasi 3D PCA
figure; % Buat jendela gambar baru.
if komponen >= 3
   W3D = W(:,1:3); % Jika ada minimal 3 komponen PCA, ambil 3 komponen pertama
untuk visualisasi 3D.
else
   W3D = [W, zeros(size(W,1), 3-komponen)]; % Jika kurang dari 3 komponen, tambahkan
kolom nol agar data bisa dipetakan ke 3D.
```

```
end

uniqueLabels = unique(labels); % Dapatkan daftar unik label jenis tanah.
colors = lines(length(uniqueLabels)); % Buat set warna berbeda untuk setiap jenis tanah.

hold on; % Izinkan banyak plot di satu gambar.
for i = 1:length(uniqueLabels) % Loop untuk setiap jenis tanah.
   idx = labels == uniqueLabels(i); % Temukan indeks gambar yang termasuk jenis tanah
ini.
   scatter3(W3D(idx,1), W3D(idx,2), W3D(idx,3), 36, colors(i,:), 'filled'); % Gambar titik-titik di
3D untuk jenis tanah ini.
end
hold off; % Hentikan penambahan plot.

title('PCA 3D Scatter of Soil Types'); % Atur judul plot.
xlabel('PC1'); % Label sumbu X sebagai Komponen Utama 1.
ylabel('PC2'); % Label sumbu Y sebagai Komponen Utama 2.
zlabel('PC3'); % Label sumbu Z sebagai Komponen Utama 3.
legend(cellstr(uniqueLabels), 'Location', 'bestoutside'); % Tampilkan legenda dengan nama
jenis tanah.
grid on; % Aktifkan grid pada plot.
view(45, 25); % Atur sudut pandang 3D agar terlihat miring.
```

3.6 Klasifikasi

Setelah data diproyeksikan ke dalam ruang berdimensi rendah menggunakan metode PCA, langkah selanjutnya adalah melakukan klasifikasi jenis tanah berdasarkan fitur hasil reduksi tersebut. Metode klasifikasi yang digunakan dalam penelitian ini adalah *K-Nearest* Neighbor (KNN), sebuah metode non-parametrik yang sederhana namun efektif. Prinsip kerja KNN adalah menentukan kelas suatu dari uji berdasarkan mayoritas kelas dari k tetangga terdekatnya di dalam ruang fitur.

Dalam implementasinya, dataset hasil reduksi PCA dibagi menjadi dua bagian menggunakan teknik *hold-out-cross-validation*,

yaitu 70% sebagai data pelatihan dan 30% sebagai data pengujian. Proses ini dilakukan dengan fungsi ***cvpartition*** pada MATLAB. Model KNN kemudian dibangun dengan ***fitcknn*** menggunakan jumlah tetangga terdekat sebanyak 3 (*NumNeighbor* = 3). Potongan kode yang menggambarkan proses pelatihan dan prediksi dengan KNN sebagai berikut:

```
% Step 4: Klasifikasi KNN
cv = cvpartition(labels, 'HoldOut', 0.3); % Bagi data menjadi set training (70%) dan testing
(30%).
XTrain = W(training(cv), :); % Ambil fitur untuk data training.
YTrain = labels(training(cv)); % Ambil label untuk data training.
XTest = W(test(cv), :); % Ambil fitur untuk data testing.
YTest = labels(test(cv)); % Ambil label untuk data testing.

knnModel = fitcknn(XTrain, YTrain, 'NumNeighbors', 3); % Buat model KNN dengan 3
tetangga terdekat.
YPred = predict(knnModel, XTest); % Prediksi label untuk data testing.
```

NumNeighbors merupakan salah satu contoh optimalisasi *hyperparameter* pada fungsi fitcknn. *NumNeighbors* digunakan dalam menentukan banyaknya jumlah tetangga terdekat atau biasa disimbolkan dengan K. Nilai K ini adalah integer positif. Secara *default*, K memiliki nilai 1 [15].

Seletah proses klasifikasi dilakukan, model KNN dievaluasi dengan menghitung confusion matrix untuk mengetahui jumlah prediksi yang benar dan salah pada masing-masing kelas tanah. Dari cofucion matrix tersebut, diperoleh akurasi klasifikasi dengan membagi jumlah prediksi yang benar (jumlah diagonal utama)

terhadap total seluruh prediksi. Selain itu, hasil cofusion matrix juga divisualisasi dalam bentuk grafik untuk memberikan gambaran performa klasifikasi secara lebih intuitif. Potongan kode evaluasi dan visualisasi hasil klasifikasi sebagai berikut:

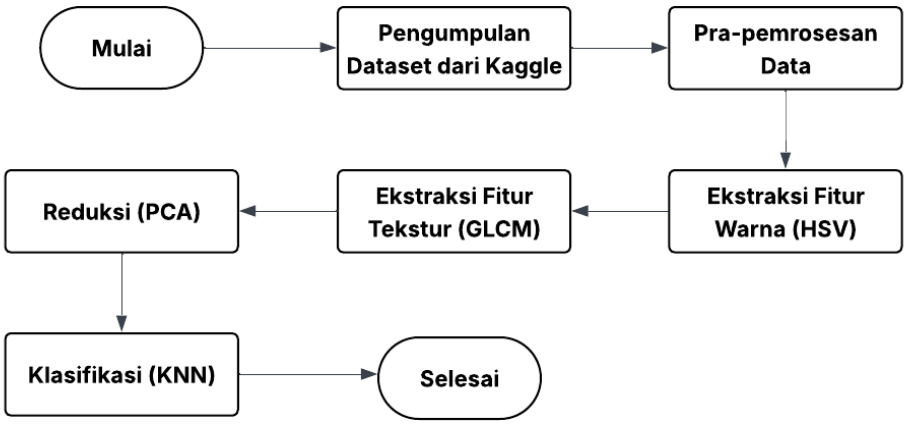
```
% Step 5: Evaluasi
confMat = confusionmat(YTest, YPred); % Buat confusion matrix.
akurasi = sum(diag(confMat)) / sum(confMat(:)) * 100; % Hitung akurasi.
fprintf('Akurasi klasifikasi: %.2f%%\n', akurasi); % Tampilkan akurasi.

% Tampilkan confusion matrix sebagai chart
figure; % Buka jendela baru.
cm = confusionchart(YTest, YPred); % Tampilkan confusion matrix dalam bentuk chart.
cm.Title = sprintf('Confusion Matrix'); % Beri judul chart.
```

3.7 Diagram Alir

Diagram alir proses eksperimen merupakan representasi visual yang menggambarkan tahapan penelitian secara sistematis, mulai dari persiapan data hingga analisis hasil. Dalam penelitian ini, alur kerja dimulai dengan pengumpulan dataset citra tanah dari platform dari *kaggle*, kemudian dilakukan pre-pemrosesan data bertujuan memastika kualitas input, dilanjutka dengan ekstraksi fitur warna menggunakan HSV (*Hue*, *Saturation*, *Value*) dan fitur tekstur dengan GLCM (*Gray Level Co-occurance Matrix*). Selanjutnya, reduksi dimensi fitur dilakukan dengan PCA (*Principal Component Analysis*) untuk menyederhanakan kompleksitas data tanpa kehilangan informassi penting. Tahap akhir melibatkan klasifikasi menggunakan algoritma *K-Nearest Neighbor* (KNN) untuk mengidentifikasi jenis

tanah berdasarkan fitur yang telah diekstraksi. Gambar Diagram alir proses eksperimen dapat dilihat pada Gambar 3.7.1.



Gambar 3.7. 1 Diagram Alir Proses Eksperimen

3.7.1 Pengumpulan Dataset dari Kaggle

Pengumpulan dataset dilakukan dengan mengambil data dari situs *Kaggle*, menggunakan dataset berjudul "Soil Type" yang berisi 156 gambar tanah dalam lima kategori: *Black Soil, Cinder Soil, Laterite Soil, Peat Soil,* dan *Yellow Soil*. Dataset ini dipilih karena memiliki keberagaman jenis tanah yang relavan untuk penelitian klasifikasi citra tanah.

3.7.2 Pra-pemrosesan Data

Tahap pra-pemrosesan data meliputi pemilihan empat kategori tanah yang digunakan (Black Soil, Laterite Soil, Peat Soil, dan Yellow Soil), penghapusan gambar dengan kualitas rendah, melakukan *cropping* untuk menghilangkan bagian yang tidak relavan, serta mengubah ukuran gambar (*resize*) menjadi

seragam, yaitu 224 x 224 piksel. Untuk menambah variasi data, diterapkan proses augmentasi seperti rotasi, *flip horizontal*, dan *flip vertical*, hingga diperoleh total 429 gambar.

3.7.3 Ekstraksi Fitur Warna

Ekstraksi fitur warna dilakukan dengan mengkonversi gambar dari format RGB ke HSV menggunakan fungsi **rgb2hsv**, lalu menghitung rata-rata (*mean*) dan standar devisiasi (*standard deviation*) dari masing-masing channel *Hue*, *Saturation*, dan *Value*. Dengan ini, diperoleh enam fitur warna dari setiap gambar.

3.7.4 Ekstraksi Fitur Tekstur

Fitur tekstur diekstraksi menggunakan metode GLCM setelah gambar dikonversi ke fromat *grayscale*. Empat properti tekstur utama, yaitu *contrast*, *correlation*, *energy*, dan *homogeneity*. Lalu dihitung dari GLCM pada empat arah (0°, 45°, 90°, dan 135°), kemudian dirata-ratakan untuk mewakili tekstur gambar.

3.7.5 Reduksi

Agar data fitur yang dihasilkan lebih sederhana dan efisien, dilakukan reduksi dimensi menggunakan metode PCA. Sebelum PCA, fitur distandardisasi terlebih dahulu. Komponen utama dipilih hingga mewakili setidaknya 90% variasi data,

kemudian data diproyeksikan ke ruang berdimensi lebih rendah, yang juga divisualisasikan dalam bentuk *scatter plot* 3D.

3.7.6 Klasifikasi

Selanjutnya melakukan proses klasifikasi menggunakan algoritma KNN dengan parameter 3 tetangga terdekat. Dataset dibagi menggunakan teknik *hold-out-cross-validation* (70% data pelatihan, 30% data pengujian). Model dievaluasi dengan *confusion matrix* dan dihitung akurasinya, sehingga diperoleh gambaran performa model dalam mengklasifikasikan jenis tanah.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pelatihan dan Pengujian Eksperimen

Sebelum model dapa melakukan proses klasifikasi, data yang tersedia perlu di proses dan dipersiapkan melalui beberapa tajapan. Salah satu tahapan penting selama proses ini adalah pembagian data latih dan data uji.

4.1.1 Persiapan Dataset

Eksperimen dilakukan dengan menggunakan MATLAB dengan menjalankan *script* yang telah disusun berdasarkan tahapan-tahapan metode. Dataset yang digunakan diperoleh dari situs *Kaggle* yang berisi kumpulan gambar jenis tanah. Namun hanya beberapa jenis tanah yang dipilih untuk eksperimen ini, yaitu *black soil*, *laterite soil*, *yellow soil*, dan *peat soil*. Setiap jenis tanah kemudian disimpan dalam folder terpisah, sehingga label dari setiap gambar dapat ditentukan berdasarkan nama folder tersebut dengan menggunakan parameter *LabelSource* pada fungsi *imageDatastore*.

4.1.2 Ekstraksi Fitur

Kemudian dilakukan ekstraksi fitur terhadap seluruh gambar dalam dataset. Terdapat dua jenis fitur utama yang diekstraksi yaitu fitur warna dan fitur tekstur. Fitur warna direpresentasikan dalam ruang warna HSV (Hue, Saturation, Value). Untuk masing-masing komponen (H, S, dan V) dihitung nilai mean dan standar deviasinya, sehingga terdapat hasil total 6 fitur.

Untuk fitur tekstur diperoleh dengan menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM). Perhitungan GLCM dilakukan pada gambar yang telah dikonversi menjadi format grayscale. GLCM diekstraksi dalam 4 arat yaitu 0, 45, 90, dan 135. Kemudian rata-rata dihitung untuk 4 fitur, yaitu *Contrast, Correlation, Energy, dan Homogeneity*. Sehingga secara keseluruhan didapatkan 10 fitur (6 dari HSV, dan 4 dari GLCM).

4.1.3 Reduksi Dimensi

Setelah proses ekstraksi fitur selesai, reduksi dimensi dilakukan pada seluruh fitur menggunakan *Principal Component Analysis* (PCA). Langkah ini bertujuan untuk menyederhanakan jumlah fitur tanpa mengurangi informasi penting secara signifikan, dan juga mempercepat proses perhitungan. Proses PCA dimulai dengan menstandarisasi seluruh fitur agar memiliki

skala yang sama, kemudian menghitung matriks kovarians dan eigen decomposition. Setelah diperoleh nilai eigen dan eigenvector, dipilih sejumlah komponen utama yang dapat mempertahankan akumulasi proposi variansi sebesar 90%.

4.1.4 Pembagian Data Latih dan Data Uji

Data yang telah melalui proses reduksi dimensi kemudian dibagi menjadi 2 bagian, yakni data latih dan data uji. Pembagian ini dilakukan menggunakan fungsi *cvpartition* pada MATLAB dengan metode *holdout*. Pembagian dilakukan dengan proporsi sebesar 70% untuk data latih dan 30% untuk data uji. Data latih digunakan sebagai referensi untuk melatih model selama proses klasifikasi, sedangkan data uji digunakan untuk mengevaluasi performa model terhadap data yang belum pernah dilihat sebelumnya.

4.1.5 Algoritma K-Nearest Neighbors (KNN)

Setelah melalui proses pembagian data latih dan data uji, maka proses penerapan algoritma K-Nearest Neighbors (KNN) dilakukan. Algoritma KNN akan menghitung jarak antara data uji dengan semua data latih menggunakan jarak Euclidean. Dengan menggunakan fungsi *fitcknn* di MATLAB, algoritma akan mencari data latih berjumlah K yang memiliki kemiripan paling tinggi dengan data uji berdasarkan nilai jaraknya, dimana nilai K pada

eksperimen ini adalah 3. Untuk setiap data uji, KNN akan mengambil 3 data latih terdekat sebagai acuan. Setelah ketiga tetangga terdekat itu ditemukan, masing-masing akan membawa label jenis tanahnya masing-masing. Label yang paling sering muncul di antara ketiganya akan menjadi hasil prediksi untuk data uji tersebut.

4.2 Hasil Eksperimen

Setelah eksperimen dijalankan, didapatkan sejumlah hasil yang dapat menunjukkan kualitas dan efektivitas model dalam pengklasifikasian jenis tanah berdasarkan gambar. Hasil-hasil berikut dapat digunakan sebagai bahan evaluasi model dalam kemampuan mengenali pola dari data.

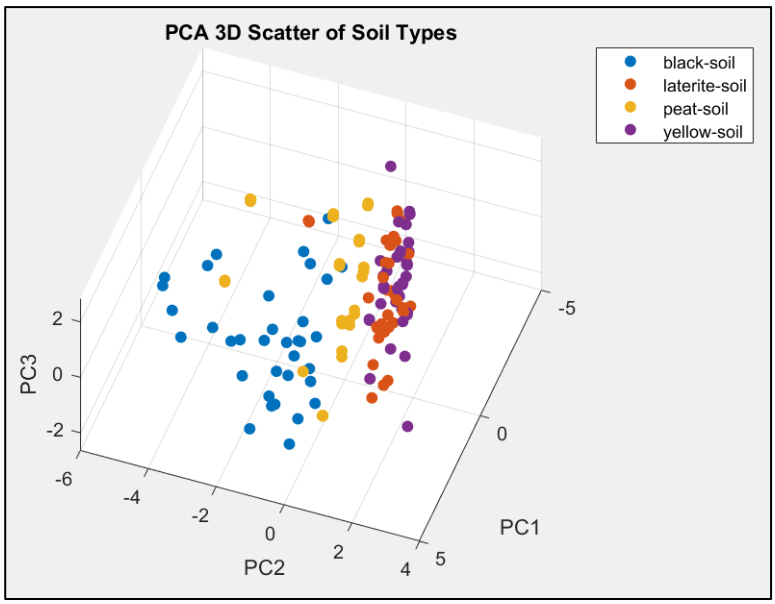
4.2.1 Jumlah Komponen Utama Hasil PCA

Setelah dilakukan ekstraksi fitur warna dalam ruang warna HSV sebanyak 6 fitur (mean dan standar deviasi dari masing-masing H, S, dan V), dan fitur tekstur menggunakan GLCM sebanyak 4 fitur (*Contrast, Correlation, Energy, dan Homogeneity*), total terdapat 10 fitur dari setiap gambar. Proses reduksi PCA dilakukan dengan tetap mempertahankan proporsi akumulasi variansi sebesar 90%. Setelah *script* dijalankan pada MATLAB, didapatkan jumlah komponen utama yang cukup untuk

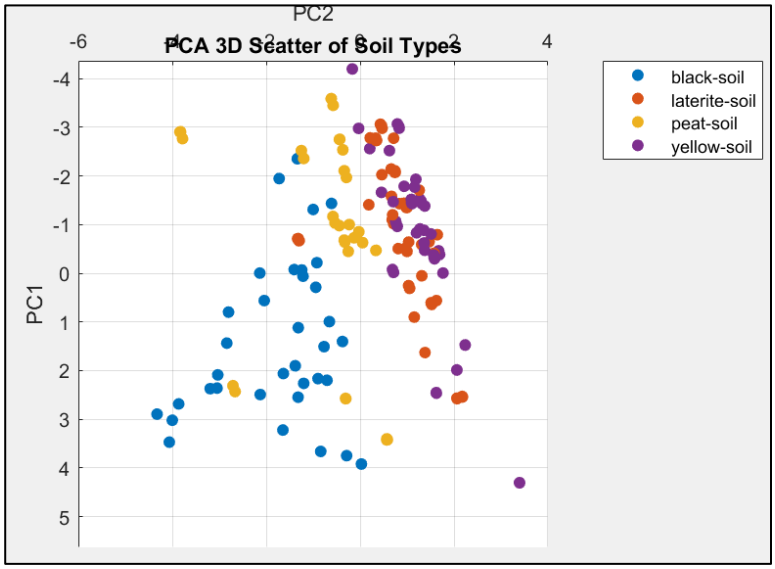
merepresentasikan 90% informasi dari keseluruhan data adalah sebanyak 5 komponen.

4.2.2 Visualisasi Sebaran Data dengan PCA 3D

Untuk memberikan gambaran distribusi data pada ruang berdimensi lebih rendah, dilakukan proyeksi ke ruang PCA 3 dimensi. Scatter plot 3D pada gambar 4.2.2.1 memperlihatkan sebaran data dari keempat jenis tanah (*black soil, laterite soil, peat soil, yellow soil*) berdasarkan 3 komponen utama pertama hasil PCA.



Gambar 4.2.2. 1 Scatter PCA 3D



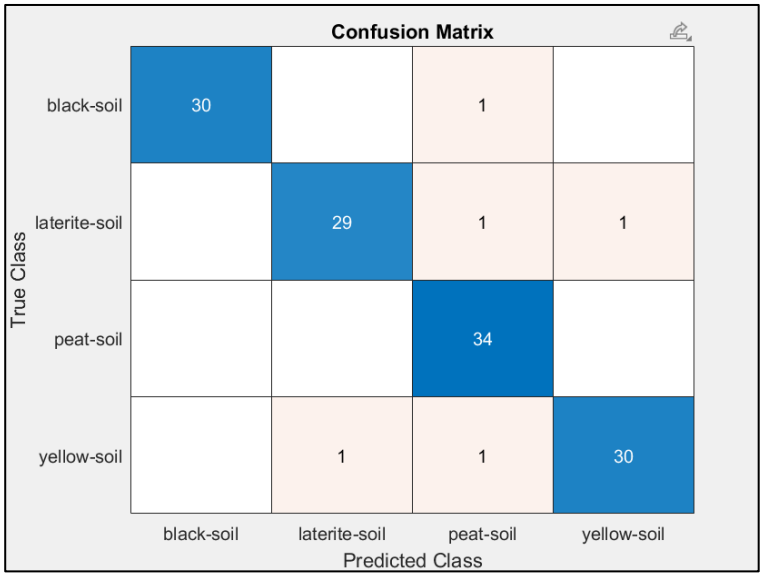
Gambar 4.2.2. 2 Scatter PCA 3D (Tampak Atas)

Dari visualisasi yang telah ditampilkan, dapat dilihat bahwa meskipun terdapat beberapa titik yang tumpang dindih antar kelas, masing-masing jenis tanah menunjukkan kecenderungan untuk membentuk kelompok yang relatif terpisah di ruang PCA. Hal ini menunjukkan bahwa fitur hasil ekstraksi mampu merepresentasikan karakteristik visual dari setiap jenis tanah, dan PCA cukup efektif dalam merangkum informasi penting dari fitur-fitur tersebut.

4.2.3 Confusion Matrix

Hitungan akurasi prediksi model menggunakan data uji sebanyak 30% dari total data. Proses klasifikasi menggunakan algoritma *K-Nearest Neighbors* (KNN) dengan nilai K sebesar 3, artinya diambil 3 tetangga yang paling dekat dengan data latih untuk menentukan label data yang diuji. Hasil prediksi model

kemudian dibandingkan dengan label asli untuk dibuat *confusion matrix* yang dapat dilihat pada gambar 4.2.3.1.



Gambar 4.2.3. 1 Confusion Matrix

Dari hasil *confusion matrix* pengujian eksperimen ini, dapat dilihat bahwa sebagian besar prediksi model sesuai dengan label aslinya, dengan kesalahan minor. Misalnya 1 sampel *black soil* disalah klasifikasikan sebagai *peat soil*, dan 1 sampel *laterite soil* disalah klasifikasikan sebagai *yellow soil*. Namun secara umum dapat dilihat bahwa ketepatan prediksi model sangat tinggi. Diperoleh juga akurasi klasifikasi model adalah sebesar 96,09%.

4.3 Analisis dan Pembahasan

Analisis dan pembahasan hasil dilakukan untuk meningkatkan pemahaman mengenai hasil-hasil yang didapatkan setelah proses pengujian eksperimen. Dengan melakukan analisis terhadap hasil, diperoleh pemahaman mengenai bagaimana kemampuan model dalam melakukan klasifikasi jenis tanah berdasarkan gambar.

4.3.1 Efektivitas Reduksi Dimensi PCA

Penerapan *Principal Component Analysis* (PCA) menghasilkan 5 komponen utama yang berhasil mempertahankan 90% akumulasi variansi dari 10 fitur awal. Hal ini menunjukkan bahwa separuh dari fitur yang awalnya digunakan menjadi tidak perlu dipertahankan seluruhnya untuk mencapai representasi yang baik terhadap data. Reduksi dimensi ini memberikan manfaat-manfaat sebagai berikut:

1. Mengurangi kompleksitas model

   Dengan hanya menggunakan 5 komponen dibandingkan 10, sehingga waktu pelatihan dapat lebih cepat dan lebih fokus ke pola-pola yang penting saja.

2. Meningkatkan visualisasi

   Dengan memanfaatkan 3 komponen pertama (PC1, PC, PC3) untuk keperluan visualisasi grafik scatter 3 dimensi, visualisasi ini mempermudah pengamatan penyebaran data

serta sebaik apa jenis-jenis tanah dapat diklasifikasikan berdasarkan karakteristiknya.

4.3.2 Sebaran Data pada Ruang PCA 3D

Visualisasi PCA dalam bentuk scatter 3D memperlihatkan gambaran awal mengenai pola keterpisahan antar jenis tanah. Meskipun terlihat pada scatter plot 3D bahwa terdapat tumpang tindih antar kelas, beberapa pola keterpisahan dapat diidentifikasi. Misalnya:

1. *Black soil* membentuk kelompok yang relative terpisah dan tidak terlalu bercampur dengan titik-titik dari kelas lainnya, artinya fitur-fitur dari black soil cukup konsisten dan berbeda dibandingkan jenis tanah lainnya.
2. *Peat soil* menunjukkan penyebaran yang lebih dibandingkan dengan black soil. Meskipun tidak terkonsentrasi di satu titik, posisi titik-titik peat soil tersebut dapat dibedakan dari jenis tanah lainnnya di ruang fitur PCA.
3. *Yellow soil* dan *Laterite soil* tampak memiliki pola titik-titik yang lebih berdekatan satu sama lain. Hal ini menunjukkan bahwa fitur yang dimiliki jenis yellow soill dan laterite soil memiliki kemiripan.

4.3.3 Akurasi Model

Berdasarkan hasil pengujian pada eksperimen ini, akurasi klasifikasi model mencapai 96,09%, dimana angka tersebut tergolong tinggi dan sangat baik. Hal ini menunjukkan bahwa kombinasi fitur warna dan tekstur tanah dapat cukup mampu mengklasifikasikan jenis tanah. Dapat dilihat *confusion matrix* hasil percobaan ini (Gambar 4.2.3.1), bahwa seluruh data uji untuk kelas peat soil berhasil diklasifikasikan dengan benar. Selain itu, dapat dilihat bahwa kesalahan klasifikasi pada kelas lain juga berjumlah kecil. Distribusi prediksi terlihat cukup seimbang dan mengindikasikan tidak adanya bias klasifikasi terhadap salah satu kelas.

Tingginya akurasi pada model ini juga menunjukkan bahwa pemilihan parameter K berjumlah 3 pada KNN merupakan pilihan yang cukup baik. Nilai K yang terlalu kecil dapat membuat model terlalu sensitif terhadap *noise*, sedangkan nilai K yang terlalu besar dapat menyebabkan bias terhadap kelas mayoritas. Namun hasil yang ditunjukkan ini tidak bersifat absolut, dikarenakan penggunaan fungsi *cvpartition* dan metode *holdout* yang membagi data latih dan data uji secara acak pada setiap percobaan. Hal ini menyebabkan hasil akurasi dan *confusion matrix* dapat memiliki perbedaan kecil jika *script* dijalankan ulang. Namun dibalik hasil yang tidak absolut, terdapat beberapa

kelebihan mengenai pemakaian fungsi *cvpartition* dan metode *holdout* dalam membagi data latih dan data uji ini, antara lain:

1. Evaluasi lebih realistis

   Pembagian data latih dan data uji secara acak lebih mendekati kondisi dunia nyata, dimana model harus mampu mengklasifikasikan data yang belum pernah dilihat sebelumnya.

2. Ketahanan model

   Pembagian yang acak juga membuat model menjadi kuat dan tetap akurat meskipun ada sedikit perubahan data.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini telah berhasil mengembangkan dan menguji sebuah sistem untuk klasifikasi jenis tanah berbasis citra digital sebagai jawaban atas tantangan di sektor pertanian. Melalui serangkaian metode yang sistematis, sistem ini dirancang secara spesifik untuk mengidentifikasi empat jenis tanah yang relevan bagi kondisi agrikultur di Indonesiaa, yaitu *Black Soil, Laterite Soil, Peat Soil,* dan *Yellow Soil*.

Alur kerja yang diterapkan, mulai dari ekstraksi fitur warna menggunakan *Hue Saturation Value* (HSV), dan fitur tekstur dengan *Gray Level Co-occurrence Matrix* (GLCM), terbukti efektif dalam menangkap karakteristik visual yang essensial untuk membedakan setiap jenis tanah. Effisiensi model secara signifikat ditingkatkan melalui implementasi *Principal Component Analysis* (PCA), yang berhasil mereduksi 10 fitur awal menjadi 5 komponen utama dengan tetap mempertahankan 90% informasi penting. Reduksi dimensi ini tidak hanya menyederhanakan komputasi tetapi juga mempertajam fokus pada fitur-fitur yang paling berpengaruh.

Puncak dari alur kerja ini adalah tahapan klassifikasi menggunakan algoritma K-*Nearest Neighbor* (KNN), yang mampu mencapai tingkat akurasi sangat tinggi sebesar 96,09%. Tingginya akurasi yang

dikonfirmasi oleh *Confusion matrix* membuktikan validitas pendekatan yang digunakan dan menunjukkan bahwa sistem yang dibangun memiliki kapabilitas yang andal untuk mengenali jenis tanah secara akurat. Secara keseluruhan, penelitian ini berhasil dikembangkan menjadi aplikasi praktis yang dapat membantu petani dalam mengambil keputusan penanaman yang lebih tepat sasaran.

5.2 Saran

Berdasarkan hasil penelitian ini mengenai penerapan metode HSV, GLCM, PCA, dan KNN untuk klasifikasi citra tanah dengan akurasi yang menjanjikan, berikut adalah beberapa saran pengembangan yang dapat dipertimbangkan untuk proyek-proyek di masa mendatang. Saran-saran ini bertujuan untuk memperluas fungsionalitas, meningkatkan akurasi, dan menjadikan sistem lebih aplikatif serta mudah diakses oleh pengguna:

5.2.1 Pengembangan Sistem Terintegrasi dengan Database Kesesuaian Tanaman

Aspek paling krusial dan memiliki dampak signifikat untuk pengembangan selanjutnya adalah integrasi sistem klasifikasi citra tanah ini dengan database komprehensif mengenai kesesuaian berbagai jenis tanaman (misalnya: "*Black Soil*", "*Laterite Soil*", "*Peat Soil*", atau "*Yellow Soil*"), secara otomatis

sistem akan menampilkan daftar rekomendasi tanaman yang paling cocok untuk jenis tanah tersebut

- **Struktur Database**: Database ini dapat mencakup berbagai parameter penting seperti jenis tanaman (padi, jagung, kedelai, sayuran, buah-buahan, dll.), preferensi jenis tanah (pH, tekstur, kandungan organik), kebutuhan air, toleransi terhadap kondisi ekstrem (kekeringan/banjir), serta potensi hasil panen pada jenis tanah tertentu.
- **Mekanisme Integrasi**: integrasi dapat dilakukan dengan membuat modul penghubung antara hasil klasifikasi KNN dengan database untuk menampilkan tanaman-tanaman yang sesuai.
- **Manfaat**: integrasi ini akan mengubah sistem dari sekedar alat klasifikasi menjadi solusi praktis yang dapat langsung memberikan rekomendasi kepada pengguna, sangat mempercepat proses pengambilan keputusan dalam pertanian, serta membantu petani dalam merencanakan penanaman yang optimal sesuai dengan kondisi tanah mereka. Ini juga akan meminimalkan risiko gagal panen akibat ketidaksesuaikan tanaman dengan jenis tanah.

5.2.2 Pengembangan Antarmuka Pengguna (UI) dan Pengalaman Pengguna (UX) yang Intuitif

Sistem dapat diadopsi secara luas oleh petani atau pihak yang tidak memilik latar belakang teknis, pengembangan antarmuka yang ramah pengguna adalah suatu keharusan;

- **Aplikasi Berbasis Web/Mobile**: sistem dapat dikembangkan menjadi aplikasi berbasis web atau mobile, memungkinkan pengguna untuk dengan mudah mengunggah foto tanah dari perangkat mereka dan mendapatkan hasil klasifikasi serta rekomendasi tanaman secara instan.
- **Visualisasi Hasil yang Jelas:** tampilkan hasil klasifikasi dengan jelas, misalnya dengan persentase probabilitas untuk setiap jenis tanah yang mungkin, serta visualisasi fitur yang telah diekstraksi jika memungkinkan.
- **Panduan Interaktif**: Sediakan panduan langkah demi langkah tentang cara mengambil foto tanah yang baik untuk analisis, serta penjelasan sederhana tentang arti dari hasil klasifikasi

Dengan menerapkan saran-saran ini, diharapkan penelitian klasifikasi citra tanah ini dapat berkembang menjadi sistem yang lebih

canggih, akurat, dan memiliki nilai aplikasi yang tinggi dalam mendukung sektor pertanian.

DAFTAR PUSTAKA

[1] K. Salma, “ANALISIS FAKTOR-FAKTOR YANG MEMPENGARUHI PRODUKSI KOPI DI INDONESIA,” *Jurnal Ilmu Sosial, Manjemen, dan Akuntansi (JISMA),* vol. 1, no. 6, pp. 805-812, 6 Februari 2022.

[2] N. A. Safitri , E. Atthoillah, A. N. Najamuddin, H. A. Fauzi, L. Hikmah, D. A. Prasetyo dan Z. E. Fitri, “Sistem Evaluasi Lahan Penentuan Tanaman menggunakan Metode Forward Chaning,” *MIND (Multimedia Artificial Intelligent Networking Database),* vol. 9, no. 2, pp. 153 - 195, 2024.

[3] A. Hadinegoro dan D. A. Rizaldhi, “Pengaruh HSV Pada Pengolahan Citra Untuk Kematangan Buah Cabai,” *Building of Informatics, Technology and Science (BITS),* vol. 3, no. 3, pp. 155-163, 12 2021.

[4] M. R. Aohana, R. N. L. Hidhayah, M. J. Andara, N. Amara dan F. Bimantoro, “Review Komprehensif: Ekstraksi Fitur GLCM, GLRLM, dan LBP untuk Pendeteksian Korosi,” vol. 3, no. 1, 13 Januari 2024.

[5] P. Putra, A. M. H. Pardede dan S. Syahputra, “ANALISIS METODE K-NEAREST NEIGHBOUR (KNN) DALAM KLASIFIKASI DATA IRIS BUNGA,” *Jurnal Teknik Informatika Kaputama (JTIK),* vol. 6, no. 1, 1 2022.

[6] R. Dijaya dan H. Setiawan, BUKU AJAR PENGOLAHAN CITRA DIGITAL, Mojopahit: UMSIDA PRESS, 2023.

[7] M. Fatwa, R. Ristu, S. Pandiangan dan E. Supriyadi, “PENGAPLIKASIAN MATLAB PADA PERHITUNGAN MATRIKS,” *Papanda Journal of Mathematics and Sciences Research (PJMSR),* vol. 1, no. 2, 12 2022.

[8] R. Pramudiya, C. Asyraq, A. Kadafi dan R. P. Sardika, “ANALISIS GAMBAR MENGGUNAKAN METODE GRAYSCALE DAN HSV (HUE, SATURATION, VALUE),” *Just IT: Jurnal Sistem Informasi, Teknologi Informasi dan Komputer,* vol. 14, no. 3, pp. 150-233, Mei 2024.

[9] M. R. Saputra, A. S. R. Anosori dan R. E. Saputra, “DETEKSI KULIT MANUSIA PADA GAMBAR MENGGUNAKAN ALGORITMA RGB DAN HSV,” *e-Proceeding of Enginnering,* vol. 8, no. 1, pp. 484-491, 2021.

[10] H. A. Rabbani, M. A. Rahman dan B. Rahayudi, “Perbandingan Ruang Warna RGB dan HSV dalam Klasifikasi Kematangan Biji Kopi,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* vol. 5, no. 6, pp. 2243-2248, 2021.

[11] R. Ikhlasul, A. Wijaya, N. D. M. Veronica dan R. Toyib, “Klasifikasi Kualitas Buah SawitMenggunakan Metode Gray Level Co-occurrence

Matrix Dengan Variasi Arah Obyek," *Jurnal Medis Infortama,* vol. 20, no. 1, pp. 256-263, 2024.

[12] L. Hakim, S. P. Kristanto, D. Yusuf dan F. N. Afia, "PENGENALAN MOTID BATIK BANYUNWANGI BERDASARKAN FITUR GRAY LEVEL CO-OCCURRENCE MATRIX," *Jurnal Teknoinfo,* vol. 16, no. 1, 2022.

[13] M. I. Al-Arrafi dan A. Ramadhanu, "IMPLEMENTASI METODE ALGORITMA PRINCIPAL COMPONENT ANALYSIS (PCA) DAN ALGORITMA K-NEAREST NEIGHBOR (KNN) DALAM KLASIFIKASI BUAH JAMBU MADU JAMBU MERAH DAN MANGGIS," *Journal of Science and Social Research,* vol. 8, no. 1, pp. 817-822, Feb 2025.

[14] MATLAB Tutorial, Tutorial Point (I) Pvt. Ltd., 2014.

[15] I. Werdiningsih, D. C. R. Novitasari dan D. Z. Haq, "Pengelolaan Data Mining Dengan Pemrograman MATLAB," Airlangga University Press, 2022.

LAMPIRAN

Kode Matlab:

```
clear; clc; close all;

% Step 1: Load image dataset
imds = imageDatastore('soil-types', ... % Membuat sebuah datastore gambar dari folder 'soil-types'
    'IncludeSubfolders', true, ...     % Menyertakan semua subfolder di dalam 'soil-types'
    'LabelSource', 'foldernames');     % Secara otomatis memberi label pada gambar berdasarkan
nama foldernya

% Step 2: Ekstraksi fitur warna (HSV) dan tekstur (GLCM)
numImages = numel(imds.Files); %numImages memuat jumlah gambar di dataset
features = []; %array kosong untuk menyimpan fitur
labels = []; %array kosong untuk label jenis tanah

% Memulai loop untuk setiap gambar
for i = 1:numImages
    img = readimage(imds, i); % membaca gambar ke i, dimasukkan ke img
    %HSV
    hsvImg = rgb2hsv(img); %konversi gambar menjadi ruang warna HSV
    % Kemudian mengambil nilai rata dan standar deviasi dari masing masing H, S, dan V
    % Sehingga total terdapat 6 fitur
    H = hsvImg(:,:,1); S = hsvImg(:,:,2); V = hsvImg(:,:,3); % ambil semua baris (:), ambil semua kolom
(:), ambil channel ke (1-2-3)
    hsvFeatures = [mean(H(:)), std(H(:)), ...
                   mean(S(:)), std(S(:)), ...
                   mean(V(:)), std(V(:))];

    %GLCM
    grayImg = rgb2gray(img); %konversi gambar ke grayscale
    % Hitung tekstur dengan melihat pasangan piksel dalam 4 arah
    % 0 1 piksel di kanan (0 derajat)
    % -1 1 diagonal kanan atas (45 derajat)
    % -1 0 piksel di atas (90 derajat)
    % -1 -1 diagonal kiri atas (135 derajat)
    glcm = graycomatrix(grayImg, 'Offset', [0 1; -1 1; -1 0; -1 -1]);
    stats = graycoprops(glcm, {'Contrast', 'Correlation', 'Energy', 'Homogeneity'}); % Hitung properti
tekstur GLCM (Contrast, Correlation, Energy, Homogeneity) dari matriks glcm.
    % mengapa glcm berukuran 8x8x4?
    % graycomatrix mengatur nilai ke 8 level intensitas keabuan
    % sehingga glcm mencatat pasangan dari level 1-8 ke 1-8 (total 8x8 kemungkinan)
    % 4 berasal dari jumlah arah (offset)
    % sehingga glcm digitung dari 4 arah berbeda
```

```
    % Ambil rata-rata dari 4 arah GLCM
    contrast = mean(stats.Contrast); % Rata-rata nilai Contrast dari semua arah.
    correlation = mean(stats.Correlation); % Rata-rata nilai Correlation dari semua arah.
    energy = mean(stats.Energy); % Rata-rata nilai Energy dari semua arah.
    homogeneity = mean(stats.Homogeneity); % Rata-rata nilai Homogeneity dari semua arah.
    glcmFeatures = [contrast, correlation, energy, homogeneity]; % Gabungkan rata-rata properti
GLCM menjadi satu array fitur.

    % Gabungkan fitur
    totalFeatures = [hsvFeatures, glcmFeatures]; % Menggabungkan fitur HSV dan GLCM menjadi
satu array fitur tunggal untuk gambar saat ini.
    features = [features; totalFeatures]; % Menambahkan array 'totalFeatures' (fitur dari gambar saat
ini) sebagai baris baru ke array 'features' yang menyimpan semua fitur dari dataset.
    labels = [labels; imds.Labels(i)]; % Menambahkan label dari gambar saat ini ke array 'labels' yang
menyimpan semua label.
end

% Step 3: PCA (standarisasi fitur HSV + GLCM)
Z = features - mean(features);        % Kurangi mean tiap fitur
stddev = std(Z);                      % Hitung standar deviasi tiap fitur
Zr = Z ./ stddev;                     % Standardisasi (bagi dengan std dev)

% Matriks kovarians dan eigen decomposition
Zcov = cov(Zr); % Hitung matriks kovarians dari data terstandardisasi (Zr).
[V, D] = eig(Zcov); % Lakukan dekomposisi eigen untuk mendapatkan vektor eigen (V) dan nilai
eigen (D).

% Urutkan eigenvalue dan eigenvector dari terbesar ke terkecil
p = size(features, 2); % Jumlah fitur.
m = 1; % Indeks untuk pengurutan.
for i = p:-1:1
    DA_urut(m,m) = D(i,i); % Urutkan eigenvalue menurun.
    VA_urut(:,m) = V(:,i); % Urutkan eigenvector sesuai eigenvalue.
    m = m + 1;
end

% Hitung proporsi varians dan pilih komponen utama yang total 90%
for i = 1:p
    eValue(i,1) = DA_urut(i,i); % Ambil eigenvalue terurut.
end

Total = sum(eValue); % Total varians.
PK = 0; % Kumulatif varians dijelaskan.
komponen = 0; % Jumlah komponen terpilih.
for j = 1:p
    proporsiValue(j,:) = eValue(j,1) / Total; % Proporsi varians per komponen.
```

```
    if PK < 0.9 % Jika kumulatif < 90%.
        PK = PK + proporsiValue(j,:); % Tambah kumulatif.
        komponen = komponen + 1; % Tambah komponen terpilih.
    end
end

% Proyeksi data ke ruang PCA
VA_Komponen = VA_urut(:,1:komponen); % Ambil eigenvector komponen terpilih.
W = Zr * VA_Komponen; % Proyeksikan data ke ruang PCA.

% Tambahan Visualisasi 3D PCA
figure; % Buat jendela gambar baru.
if komponen >= 3
    W3D = W(:,1:3); % Jika ada minimal 3 komponen PCA, ambil 3 komponen pertama untuk
visualisasi 3D.
else
    W3D = [W, zeros(size(W,1), 3-komponen)]; % Jika kurang dari 3 komponen, tambahkan kolom
nol agar data bisa dipetakan ke 3D.
end

uniqueLabels = unique(labels); % Dapatkan daftar unik label jenis tanah.
colors = lines(length(uniqueLabels)); % Buat set warna berbeda untuk setiap jenis tanah.

hold on; % Izinkan banyak plot di satu gambar.
for i = 1:length(uniqueLabels) % Loop untuk setiap jenis tanah.
    idx = labels == uniqueLabels(i); % Temukan indeks gambar yang termasuk jenis tanah ini.
    scatter3(W3D(idx,1), W3D(idx,2), W3D(idx,3), 36, colors(i,:), 'filled'); % Gambar titik-titik di 3D
untuk jenis tanah ini.
end
hold off; % Hentikan penambahan plot.

title('PCA 3D Scatter of Soil Types'); % Atur judul plot.
xlabel('PC1'); % Label sumbu X sebagai Komponen Utama 1.
ylabel('PC2'); % Label sumbu Y sebagai Komponen Utama 2.
zlabel('PC3'); % Label sumbu Z sebagai Komponen Utama 3.
legend(cellstr(uniqueLabels), 'Location', 'bestoutside'); % Tampilkan legenda dengan nama jenis
tanah.
grid on; % Aktifkan grid pada plot.
view(45, 25); % Atur sudut pandang 3D agar terlihat miring.

% Step 4: Klasifikasi KNN
cv = cvpartition(labels, 'HoldOut', 0.3); % Bagi data menjadi set training (70%) dan testing (30%).
XTrain = W(training(cv), :); % Ambil fitur untuk data training.
YTrain = labels(training(cv)); % Ambil label untuk data training.
XTest = W(test(cv), :); % Ambil fitur untuk data testing.
YTest = labels(test(cv)); % Ambil label untuk data testing.
```

```
knnModel = fitcknn(XTrain, YTrain, 'NumNeighbors', 3); % Buat model KNN dengan 3 tetangga
terdekat.
YPred = predict(knnModel, XTest); % Prediksi label untuk data testing.

% Step 5: Evaluasi
confMat = confusionmat(YTest, YPred); % Buat confusion matrix.
akurasi = sum(diag(confMat)) / sum(confMat(:)) * 100; % Hitung akurasi.
fprintf('Akurasi klasifikasi: %.2f%%\n', akurasi); % Tampilkan akurasi.

% Tampilkan confusion matrix sebagai chart
figure; % Buka jendela baru.
cm = confusionchart(YTest, YPred); % Tampilkan confusion matrix dalam bentuk chart.
cm.Title = sprintf('Confusion Matrix'); % Beri judul chart.
```