

Klasifikasi Kualitas Varietas Benih Jagung Bima 20 Menggunakan Metode Random Forest

Muhamad Zahara Anugrah Putra ^{1*)}, Feri Candra ²⁾, Esa Prakasa ³⁾

¹⁾²⁾ Program Studi Teknik Informatika, Fakultas Teknik, Universitas Riau

³⁾ Organisasi Riset Elektronika Dan Informatika, Badan Riset Dan Inovasi Nasional

^{*)}Correspondence author: muhamad.zahara1497@student.unri.ac.id, Riau, Indonesia

DOI: <https://doi.org/10.37012/jtik.v10i2.2177>

Abstrak

Varietas benih jagung Bima-20 merupakan salah satu varietas yang populer dan banyak digunakan oleh petani. Namun, untuk memastikan kualitas benih yang dihasilkan, diperlukan metode yang dapat membedakan kualitas benih Bima-20 dengan akurasi tinggi. Salah satu cara untuk meningkatkan akurasi dalam proses grading benih jagung adalah dengan menggunakan teknologi pengolahan citra digital. Beberapa fitur yang dapat diekstraksi dari citra digital antara lain bentuk, tekstur, dan warna. Karakteristik bentuk benih jagung dapat diekstraksi dengan menggunakan metode segmentasi citra dan ekstraksi fitur bentuk seperti area dan perimeter atau keliling. Sedangkan karakteristik tekstur benih jagung dapat diekstraksi dengan menggunakan fitur *gray-level co-occurrence matrix* (GLCM) serta dapat diklasifikasi menggunakan metode *Random Forest*. Metode *Random Forest* adalah salah satu metode yang populer dalam klasifikasi citra. Metode ini menggunakan kombinasi dari beberapa pohon keputusan (*decision tree*) untuk mengklasifikasikan data. Kelebihan dari metode *Random Forest* adalah kemampuannya dalam mengatasi *overfitting* dan mampu menghasilkan prediksi yang akurat. Dengan menerapkan ekstraksi fitur dan metode tersebut menghasilkan bahwa ekstraksi fitur tekstur menggunakan *Gray Level Co-occurrence Matrix* (GCLM) dan ekstraksi fitur bentuk memperoleh nilai yang dapat diklasifikasikan menggunakan metode *random forest*. Hasil klasifikasi yang diperoleh tersebut memiliki tingkat akurasi 100% akurat sesuai dengan pernyataan melalui survei yang dilakukan kepada seorang kepala SMK di Pesantren Teknologi Riau dan juga seorang guru dalam bidang pertanian di Pesantren Teknologi Riau yaitu ibu Azrida Syamsi M.Si.

Kata Kunci: Benih Jagung Bima 20, Pengolahan Citra Digital, Ekstraksi Fitur Tekstur, Ekstraksi Fitur Bentuk, Random Forest

Abstract

The Bima-20 corn seed variety is a popular variety and is widely used by farmers. However, to ensure the quality of the seeds produced, a method is needed that can differentiate the quality of Bima-20 seeds with high accuracy. One way to increase accuracy in the corn seed grading process is to use digital image processing technology. Some features that can be extracted from digital images include shape, texture and color. The shape characteristics of corn seeds can be extracted using image segmentation methods and extraction of shape features such as area and perimeter or circumference. Meanwhile, the texture characteristics of corn seeds can be extracted using the gray-level co-occurrence matrix (GLCM) feature and can be classified using the Random Forest method. The Random Forest method is one of the popular methods in image classification. This method uses a combination of several decision trees to classify data. The advantage of the Random Forest method is its ability to overcome overfitting and produce accurate predictions. By applying feature extraction and this method, texture feature extraction using the Gray Level Co-occurrence Matrix (GCLM) and shape feature extraction obtain values that can be classified using the random forest method. The classification results obtained have an accuracy rate of 100% according to statements made through a survey conducted on a vocational school principal at the Riau Technology Islamic Boarding School and also a teacher in agriculture at the Riau Technology Islamic Boarding School, namely Mrs. Azrida Syamsi M.Si..

Keywords: Bima 20 Corn Seeds, Digital Image Processing, Texture Feature Extraction, Shape Feature Extraction, Random Forest

PENDAHULUAN

Benih jagung merupakan faktor penting dalam keberhasilan produksi pertanian. Kualitas benih yang baik dapat meningkatkan produktivitas dan kualitas hasil panen. Oleh karena itu, penting untuk memiliki sistem yang efektif dalam mengklasifikasikan kualitas varietas benih jagung. Varietas benih jagung Bima-20 merupakan salah satu varietas yang populer dan banyak digunakan oleh petani. Namun, untuk memastikan kualitas benih yang dihasilkan, diperlukan metode yang dapat membedakan kualitas benih Bima-20 dengan akurasi tinggi.

Salah satu cara untuk meningkatkan akurasi dalam proses grading benih jagung adalah dengan menggunakan teknologi pengolahan citra digital. Pengolahan citra digital telah menjadi alat yang efektif dalam menganalisis dan memanipulasi citra. Dalam konteks ini, pengolahan citra digital dapat digunakan untuk mengklasifikasikan kualitas varietas benih jagung Bima 20 berdasarkan fitur-fitur yang ada dalam citra.

Karakteristik bentuk dan tekstur benih jagung dapat diekstraksi dari citra digital benih jagung. Beberapa fitur yang dapat diekstraksi dari citra digital antara lain bentuk, tekstur, dan warna. Karakteristik bentuk benih jagung dapat diekstraksi dengan menggunakan metode segmentasi citra dan ekstraksi fitur bentuk seperti area dan perimeter atau keliling. Sedangkan karakteristik tekstur benih jagung dapat diekstraksi dengan menggunakan fitur *Gray-Level Co-occurrence Matrix* (GLCM) serta dapat diklasifikasi menggunakan metode *Random Forest*. Metode *Random Forest* adalah salah satu metode yang populer dalam klasifikasi citra. Metode ini menggunakan kombinasi dari beberapa pohon keputusan (*decision tree*) untuk mengklasifikasikan data. Kelebihan dari metode *Random Forest* adalah kemampuannya dalam mengatasi *overfitting* dan mampu menghasilkan prediksi yang akurat.

Beberapa fitur ekstraksi yang diterapkan oleh penelitian terdahulu seperti *eccentricity* untuk fitur ekstraksi bentuk dan HSV (*Hue-Saturation-Value*) untuk fitur ekstraksi warna yang dilakukan oleh Yafie, H. A., Rachmawati, E., Prakasa, E., dan Nur, A pada tahun 2019 yang menjelaskan bahwa proses identifikasi dalam penelitian ini memiliki tiga tahapan yaitu pemilihan ROI, ekstraksi fitur dan klasifikasi menggunakan metode *Artificial Neural*

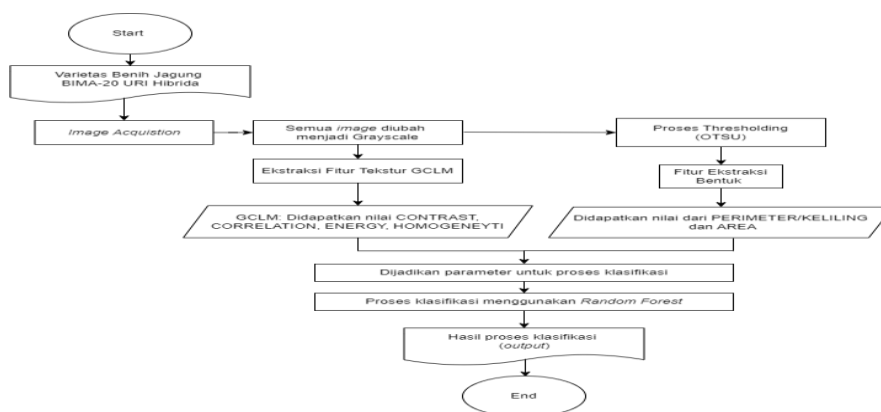
Network (ANN). Permasalahan yang terjadi pada penelitian ini yaitu pemilihan atau sortasi jagung penting untuk menghasilkan benih berkualitas tinggi sebelum didistribusikan ke daerah-daerah dengan kondisi serta karakteristik pertanian yang bervariasi. Sebab itu perlu dibangun identifikasi benih jagung, dimana penulis mengusulkan teknik identifikasi jagung yang menggabungkan keuntungan dari menggabungkan fitur bentuk *eccentricity* dan warna. Fitur warna diekstraksi berdasarkan saluran HSV (*Hue- Saturation-Value*). Hasil yang diperoleh dalam penelitian ini yaitu sistem yang diusulkan mencapai kinerja yang sangat baik untuk mengidentifikasi kualitas jagung yang buruk dan baik untuk spesies BIMA-20 dan NASA-29. Hasil untuk klarifikasi BIMA-20 baik vs. BIMA-20 buruk memberikan akurasi sebesar 89%, sedangkan akurasi klasifikasi BIMA-20 baik vs. NASA-29 Baik adalah 97%. (Yafie et al., 2020).

Berdasarkan uraian yang telah dijelaskan, pada penelitian ini yaitu membuat sebuah sistem yang mampu mengidentifikasi dan mengklasifikasi benih jagung berdasarkan karakteristik bentuk dan tekstur. Citra dari benih jagung akan diambil menggunakan kamera. Untuk tahap setelahnya yang dilakukan pada penelitian ini adalah *preprocessing*. Dengan dilakukannya penelitian ini dapat menciptakan sistem yang dapat mengidentifikasi dan mengklasifikasi kualitas benih jagung BIMA -20 yang akurat dan efisien. Kelebihan dalam penelitian ini yaitu menggunakan beberapa fitur ekstraksi yang telah diterapkan oleh peneliti terdahulu seperti *Gray-Level Co-occurrence Matrix* (GLCM) untuk tekstur pada benih jagung. Dalam penelitian kali ini penulis menambahkan satu fitur ekstraksi untuk proses karakteristik bentuk pada jagung yang menggunakan area dan perimeter atau keliling serta metode klasifikasi *Random Forest* yang memiliki beberapa kelebihan dari fitur yang telah diterapkan oleh peneliti terdahulu seperti tidak dapat membedakan antara bentuk yang serupa tetapi dengan orientasi yang berbeda. Menggunakan fitur ekstraksi ini dapat digunakan untuk mengukur tingkat kelengkungan atau kerutan pada benih jagung berdasarkan bentuknya, fitur ekstraksi tekstur ini tidak begitu bergantung pada intensitas cahaya pada citra sehingga dapat digunakan pada kondisi pencahayaan yang berbeda dan yang terakhir kelebihan metode klasifikasi ini adalah metode tersebut relatif mudah diimplementasikan pada pengklasifikasikan benih jagung. Maka dari itu peneliti membuat

suatu penelitian mengenai “Klasifikasi Kualitas Varietas Benih Jagung Bima 20 Menggunakan Metode Random Forest”.

METODE

Penulis menggunakan metode pengolahan citra digital. Metode pengolahan citra digital merupakan suatu metode yang menggunakan teknologi pengolahan citra digital untuk memperoleh data citra secara komputerisasi. Tujuan dari penerapan metode pengolahan citra digital yaitu untuk mendapatkan informasi yang lebih baik serta lebih akurat dari citra, sehingga dapat diterapkan untuk kepentingan penelitian atau aplikasi praktis. Dalam penelitian ini, alur kerja yang akan dilakukan untuk pengembangan algoritma citra digital *random forest* sebagai berikut:



Gambar 1. Algoritma Citra Digital Random Forest

Berikut adalah penjelasan mengenai alur kerja:

- Varietas benih jagung BIMA-20 URI Hibrida, merupakan varietas jenis jagung yang akan digunakan dalam penelitian ini
- Image acquisition*, adalah tahap pertama dalam proses citra digital yang merupakan proses untuk pengambilan gambar dari dunia nyata dan mengkonversikannya menjadi bentuk digital yang dapat diperlakukan oleh komputer atau software dalam memproses citra
- Semua image diubah menjadi *grayscale*, adalah tahap dimana sebuah gambar berwarna dikonversikan menjadi gambar dengan skala keabuan (*grayscale*), dimana setiap piksel hanya memiliki tingkah keabuan yang menggambarkan intensitas

- cahaya pada titik tersebut. Dalam citra tahap *grayscale* untuk menghilangkan warna dan hanya meningkatkan keabuan (skala dari hitam sampai putih) yang tetap. Setelah melakukan proses tersebut selanjutnya melakukan ekstraksi tekstur dengan GCLM dan mendapatkan nilai *contrast*, *correlation*, *energy* dan *homogeneity*
- d. Proses *thresholding* (OTSU), merupakan teknik yang digunakan dalam proses citra untuk secara otomatis menentukan nilai ambang optimal yang memisahkan piksel menjadi dua kelas berdasarkan histogram intensitas piksel. Dua kelas yang dimaksud tersebut yaitu objek (*foreground*) dan latar belakang (*background*) yang sering mewakili dalam citra. Setelah melakukan proses tersebut selanjutnya melakukan ekstraksi bentuk pada benih jagung dan hasil yang diperoleh berupa nilai dari perimeter atau keliling dan area
 - e. Dijadikan parameter untuk proses klasifikasi, setelah memperoleh nilai dari masing-masing ekstraksi yang telah dilakukan pada proses image menjadi *grayscale* dan proses *thresholding* tersebut maka nilai tersebut dijadikan parameter untuk dapat dilakukan proses klasifikasi
 - f. Proses klasifikasi menggunakan *random forest*, melakukan proses klasifikasi dengan metode yang diterapkan yaitu *random forest*
 - g. Hasil proses klasifikasi (*output*), hasil yang didapatkan setelah melakukan proses klasifikasi *random forest*

HASIL DAN PEMBAHASAN

Perisapan Kebutuhan Alat dan Bahan Penelitian

Tahap awal ini merupakan tahap persiapan yang dilakukan oleh penulis untuk dapat mempersiapkan apa saja alat dan bahan yang dibutuhkan untuk melaksanakan penelitian. Berikut adalah alat dan bahan yang diperlukan :

1. Laptop dengan spesifikasi *processor*: Intel® Celeron® N4020 *Processor* 1.1 GHz (4M Cache, up to 2.8 GHz), *memory*: 4GB DDR4 SO-DIMM

2. Dataset yang digunakan dari dataset yang disimpan oleh pihak Badan Riset dan Inovasi Nasional (BRIN) berupa dataset citra benih jagung yang berjumlah 1.417 sampel benih jagung pada 100 *image*.
3. *Software Pycharm* yang digunakan sebagai pengembangan *source code*.

Image Acquisition

Dalam penelitian ini, data gambar diperoleh dari pihak Badan Riset dan Inovasi Nasional (BRIN) berupa sampel dataset benih jagung jenis bima-20. Data tersebut merupakan data yang diperoleh dari peneliti terdahulu yang telah melakukan penelitian mengenai benih jagung oleh Haddad Alwi Yafie, Ema Rachmawati¹, Esa Prakasa dan Amin Nur pada tahun 2019. Dataset tersebut diambil menggunakan kamera DSLR Canon EOS 70D dengan pencahayaan yang diatur dengan baik agar benih jagung dapat terlihat dengan jelas. Proses pengambilan citra dilakukan di Laboratorium Kelompok Keahlian Computer Vision, Pusat Lembaga Ilmu Pengetahuan Indonesia (LIPI) Bandung. (Yafie et al., 2020) Berikut adalah beberapa sampel dataset benih jagung yang akan dijadikan bahan penelitian dalam penelitian ini:



Gambar 2. Sample Dataset Benih Jagung

Proses Ubah *Image* Menjadi *Grayscale*

Proses mengubah *image* menjadi *grayscale* adalah tahapan yang melibatkan konversi warna dari citra yang berwarna menjadi citra yang hanya menggunakan skala keabuan (*grayscale*). Setiap piksel hanya memiliki tingkat kecerahan tanpa informasi warna. Dilakukannya proses ini untuk memperoleh gambar yang memiliki skala keabuan (*grayscale*) agar dapat dilakukan proses ekstraksi fitur tekstur menggunakan *Gray Level Co-occurrence Matrix* (GLCM) dan fitur ekstraksi bentuk. Berikut tahapan yang dilakukan dalam proses ubah *image* menjadi skala keabuan (*grayscale*):

1. Import Image

Import image merupakan proses dimana citra benih jagung tersebut diimport kedalam perangkat lunak pengolahan citra atau bahasa pemrograman untuk membuka gambar atau citra yang akan diubah menjadi *grayscale*. Berikut adalah *source code* untuk *import image*:

```
# Membaca semua file gambar dalam folder
for filename in os.listdir(image_folder):
    if filename.endswith('.jpg'): # Sesuaikan ekstensi file gambar
        image_path = os.path.join(image_folder, filename)
        image_paths.append(image_path)
        # Tentukan label sesuai dengan nama file atau aturan tertentu
        labels.append(random.choice(["good", "bad"]))

label_encoder = LabelEncoder()
numerical_labels = label_encoder.fit_transform(labels)
```

Gambar 3. Source Code Import Image

2. Konversi Ke Grayscale

Setelah melakukan *Import Image* langkah selanjutnya yaitu melakukan konversi ke *grayscale*. Dalam citra *grayscale*, setiap piksel diwakili oleh satu nilai kecerahan (intensitas), yang berada dalam rentang 0 (hitam) hingga 255 (putih). Berikut adalah *source code* yang digunakan untuk melakukan proses konversi *image* yang telah diimport ke *grayscale*:

```
# Konversi citra RGB ke grayscale
grayscale_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

Gambar 4. Konversi ke Grayscale

3. Output Proses Grayscale dan Simpan Hasil

Ketika konversi berhasil dilakukan, maka akan muncul output dari *grayscale* dan data tersebut akan disimpan untuk digunakan dalam proses ekstraksi fitur tekstur menggunakan *Gray Level Co-occurrence Matrix (GLCM)* dan fitur ekstraksi bentuk. Berikut adalah salah satu hasil konversi dan *source code* yang.

```
# Save the grayscale image
save_image('corn_seed_grayscale.jpg', gray_img)
```

Gambar 5. Source Code hasil Konversi



Gambar 6. Hasil Konversi

Ekstraksi Fitur Tekstur *Gray Level Co-occurrence Matrix* (GLCM)

Ekstraksi fitur tekstur menggunakan *Gray Level Co-occurrence Matrix* (GLCM) merupakan suatu metode dalam analisis citra yang digunakan untuk mengekstraksi informasi tekstur dari citra *grayscale*. GLCM mengukur seberapa sering sepasang piksel dengan intensitas keabuan tertentu muncul secara bersamaan dengan jarak dan arah tertentu dalam citra. Proses ini membantu menggambarkan distribusi tekstur dalam suatu citra. Langkah-langkah yang dilakukan dalam ekstraksi fitur tekstur menggunakan *Gray Level Co-occurrence Matrix* (GLCM) sebagai berikut:

1. Menentukan Parameter GLCM

Proses penentuan parameter dalam GLCM yaitu penentuan arah dan jarak yang akan digunakan untuk menghitung matriks *co-occurrence*. Arah biasanya dapat menjadi 0, 45, 90 dan 135 derajat. Sedangkan jarak adalah jarak antara piksel yang dipertimbangkan (jarak dan derajat belum dipertimbangkan).

2. Fungsi Pengambilan fitur GLCM

Setelah menentukan parameter GLCM, Berikut *source code* fungsi untuk mengambil fitur glcm dari citra:

```
# Fungsi untuk mengambil fitur GLCM dari citra
def extract_glcml_features(image):
    glcm = graycomatrix(image, [1], [0], symmetric=True, normed=True)
    contrast = graycoprops(glcml, 'contrast')[0, 0]
    correlation = graycoprops(glcml, 'correlation')[0, 0]
    energy = graycoprops(glcml, 'energy')[0, 0]
    homogeneity = graycoprops(glcml, 'homogeneity')[0, 0]
    return [contrast, correlation, energy, homogeneity]
```

Gambar 7. Source Code Fungsi Untuk Mengambil Fitur Glem dari Citra

3. Hasil nilai Ekstraksi tekstur

Setelah Setelah melakukan proses ekstraksi fitur tekstur dengan menggunakan *Gray Level Co-occurrence Matrix* (GLCM) maka proses selanjutnya yaitu hasil nilai ekstraksi tekstur. Tabel berikut akan menampilkan nilai-nilai fitur tekstur yang telah diekstraksi dari beberapa sampel benih jagung:

Tabel 1. Nilai – Nilai Fitur Tekstur yang Telah Diekstraksi

Nama File	<i>Gray Level Co-occurrence Matrix (GLCM)</i>			
	<i>Contrast</i>	<i>Correlation</i>	<i>Energy</i>	<i>Homogeneity</i>
images/bad\Hasil ROI Bad (1).jpg	370,9550494	0,987161804	0,741596752	0,994295281
images/bad\Hasil ROI Bad (10).jpg	850,3512397	0,966727418	0,770751898	0,986922904
images/bad\Hasil ROI Bad (100).jpg	355,8424022	0,983224518	0,817522934	0,99452769
images/bad\Hasil ROI Bad (101).jpg	318,4337904	0,985814737	0,806165336	0,995102977
images/bad\Hasil ROI Bad (102).jpg	314,2128251	0,986498217	0,798309976	0,995167889
images/good\Hasil ROI Good (1).jpg	293,2481351	0,987403914	0,798423869	0,995490294
images/good\Hasil ROI Good (10).jpg	264,5293773	0,988553352	0,800343816	0,995931944
images/good\Hasil ROI Good (100).jpg	270,9428759	0,988747847	0,790913238	0,995833315
images/good\Hasil ROI Good (101).jpg	271,8054352	0,987762487	0,808865629	0,99582005
images/good\Hasil ROI Good (102).jpg	269,4376394	0,988133386	0,804171726	0,995856463

Proses *Thresholding* (OTSU)

Proses *thresholding* OTSU adalah proses yang dikenal sebagai metode OTSU atau *thresholding Otsu's method* yang merupakan metode otomatis untuk menentukan nilai ambang (*threshold*) yang optimal untuk memisahkan antara objek dengan latar belakang dalam suatu citra. Tujuan proses ini adalah untuk membagi piksel dalam citra menjadi dua kelas yaitu satu untuk objek dan satu untuk latar belakang, berdasarkan ambang tertentu. Selain itu, bertujuan untuk memperoleh nilai ambang yang memaksimalkan varians antar kedua kelas tersebut, sehingga memastikan bahwa kedua kelas tersebut terpisah secara optimal. Dalam proses *thresholding* terdapat beberapa cara yang perlu dilakukan, sebagai berikut:

1. Import Image

Import image adalah proses impor citra benih jagung kedalam perangkat lunak pengolahan citra atau bahasa pemrograman untuk membuka gambar atau citra yang akan diproses kedalam tahap *thresholding*. Berikut adalah *source code* untuk *import image*:

```
# Membaca semua file gambar dalam folder
for filename in os.listdir(image_folder):
    if filename.endswith(".jpg"): # Sesuaikan ekstensi file gambar
        image_path = os.path.join(image_folder, filename)
        image_paths.append(image_path)
        # Tentukan label sesuai dengan nama file atau aturan tertentu
        labels.append(random.choice(["good", "bad"]))

label_encoder = LabelEncoder()
numerical_labels = label_encoder.fit_transform(labels)
```

Gambar 8. Source Code Untuk Import Image

2. Konversi Thresholding

Setelah melakukan *import image* maka tahap selanjutnya yaitu melakukan konversi *thresholding*, berikut *source code* yang digunakan untuk mendapatkan hasil dari konversi *thresholding* tersebut:

```
# Thresholding (gunakan Otsu)
_, thresholded = cv2.threshold(gray_image, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
return thresholded
```

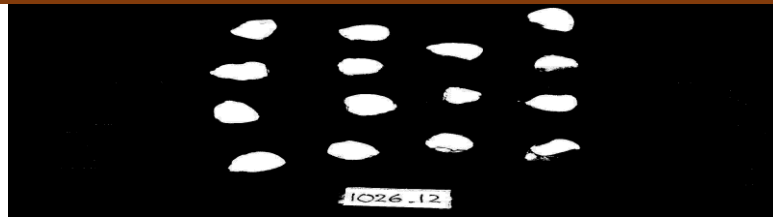
Gambar 9. Hasil dari konversi thresholding

3. Output Thresholding dan Simpan Hasil

Setelah diproses maka hasil atau output dari konversi *thresholding* akan diperoleh serta data yang telah diperoleh tersebut akan digunakan dalam proses ekstraksi fitur bentuk. Berikut adalah salah satu hasil dari konversi *thresholding* dan *source code* yang digunakan

```
input_folder = 'images'
output_folder = 'results'
threshold = 128
process_files(input_folder, output_folder, threshold)
```

Gambar 10. Source Code



Gambar 11. Hasil dari konversi *thresholding*

Proses *Region of Interest* (ROI)

Region of Interest (ROI) adalah proses identifikasi dan isolasi area tertentu atau wilayah tertentu dalam citra yang dianggap penting atau menarik untuk dianalisis lebih lanjut. ROI bertujuan untuk memfokuskan perhatian pada bagian tertentu dari citra, sehingga memungkinkan analisis yang lebih mendalam dan efisien pada area yang relevan.

Proses ROI dilakukan dengan mengupload data yang telah diproses oleh *thresholding*. Berikut adalah *source code* yang digunakan untuk memperoleh hasil ROI :

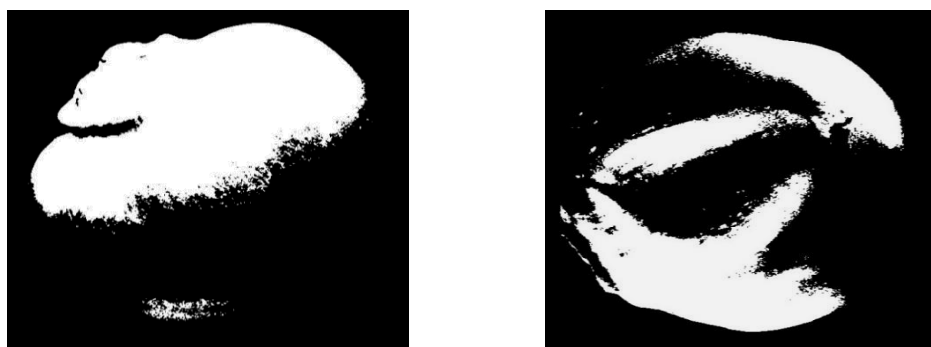
```
# Fungsi untuk mengekstrak ROI dari citra
def extract_roi(image):
    # Cari kontur pada citra
    contours, _ = cv2.findContours(image, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

    for contour in contours:
        # Asumsikan ROI berbentuk segi empat dengan warna hijau
        if cv2.contourArea(contour) > 1000:
            x, y, width, height = cv2.boundingRect(contour)
            roi = image[y:y+height, x:x+width]
            return roi

    return None # Jika ROI tidak ditemukan
```

Gambar 12. Source Code Untuk Memperoleh Hasil ROI

Dan Beberapa hasil yang diperoleh :



Gambar 13. Beberapa Hasil yang diperoleh

Ekstraksi Fitur Bentuk

Ekstraksi fitur bentuk adalah ekstraksi yang melibatkan identifikasi dan pengukuran karakteristik geometris dari suatu objek atau region dalam citra. Fitur bentuk dapat memberikan informasi tentang struktur dan karakteristik objek, dan hal tersebut digunakan dalam berbagai aplikasi seperti pengenalan pola, identifikasi objek serta segmentasi citra. Terdapat beberapa langkah dalam ekstraksi fitur bentuk, sebagai berikut:

1. Fitur-fitur Bentuk

Dalam ekstraksi fitur bentuk, terdapat dua fitur yang berbeda, yaitu *Contour Area* (Luas Kontur) dan *Contour Arc Length* (Panjang lengkungan kontur). Kedua fitur tersebut memberikan informasi yang berbeda mengenai geometri objek dalam citra. *Contour area* memberikan informasi tentang seberapa besar suatu objek, sementara *contour arc length* memberikan informasi tentang kompleksitas atau "kekakuan" kontur objek tersebut.

a. *Contour Area* (Luas Kontur)

Contour area bertujuan untuk mengukur luas daerah yang dibatasi oleh kontur suatu objek. *Contour area* diukur dalam satuan piksel persegi atau unit area yang relevan dengan resolusi dari citra. Rumus umum dari *contour area* yaitu `'cv2.contourArea(contour)'`, dimana `'contour'` adalah objek.

b. *Contour Arc Length* (Panjang Lengkungan Kontur)

Contour Arc Length mengukur panjang lengkungan atau garis yang membentuk kontur suatu objek. Pengukuran ini diukur dalam satuan piksel atau unit panjang yang relevan dengan resolusi citra. Rumus umum yang digunakan *Contour Arc Length* adalah `'cv2.arcLength(contour, closed)'` dimana `'contour'` adalah kontur objek dan `'closed'` merupakan parameter boolean yang menentukan apakah kontur tersebut tertutup atau tidak.

2. Fungsi Pengambilan Ekstraksi Fitur Bentuk

Berikut *source code* fungsi untuk mengambil fitur ekstraksi bentuk dari citra:

```
# Fungsi untuk mengekstrak ROI dari citra
def extract_roi(image):
    # Cari kontur pada citra
    contours, _ = cv2.findContours(image, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

    for contour in contours:
        # Asumsikan ROI berbentuk segi empat dengan warna hijau
        if cv2.contourArea(contour) > 1000:
            x, y, width, height = cv2.boundingRect(contour)
            roi = image[y:y+height, x:x+width]
            return roi

    return None # Jika ROI tidak ditemukan
```

Gambar 14. Source Code Fungsi Untuk Mengambil Fitur Ekstraksi Bentuk dari Citra

3. Hasil Nilai Ekstraksi Bentuk

Tabel 2. Nilai ekstraksi fitur bentuk dari beberapa sampel benih jagung:

Nama File	Fitur Ekstraksi Bentuk	
	Area	Perimeter
images/bad\Hasil_ROI_Bad (1).jpg	147638,5	1967,103802
images/bad\Hasil_ROI_Bad (10).jpg	106156,5	1701,540315
images/bad\Hasil_ROI_Bad (100).jpg	178728,5	1626,060086
images/bad\Hasil_ROI_Bad (101).jpg	186730,5	1669,859074
images/bad\Hasil_ROI_Bad (102).jpg	171957	1822,670256
images/good\Hasil_ROI_Good (1).jpg	178619,5	1761,824585
images/good\Hasil_ROI_Good (10).jpg	187840,5	1839,966721
images/good\Hasil_ROI_Good (100).jpg	192263	1701,013403
images/good\Hasil_ROI_Good (101).jpg	189437	1747,841829
images/good\Hasil_ROI_Good (102).jpg	190524	1718,243849

Proses Parameter Klasifikasi

Proses parameter klasifikasi merupakan langkah-langkah yang dilakukan untuk menentukan parameter atau konfigurasi model klasifikasi yang akan digunakan untuk memprediksi kelas dari data yang belum diketahui. Dalam konteks klasifikasi, model mengacu pada algoritma yang telah dilatih untuk memahami pola dari data pelatihan yang kemudian dapat digunakan untuk mengklasifikasikan data baru. Parameter model mengacu pada konfigurasi atau peraturan yang dapat mempengaruhi kinerja model tersebut.

1. Klasifikasi Menggunakan *Random Forest*

Proses klasifikasi menggunakan algoritma *random forest*.

a. Buat Model

Proses pembuatan model terdiri dari tiga tahap sebagai berikut:

1. *Impor* pustaka atau modul yang diperlukan, pada penelitian kali ini modul yang digunakan yaitu *matplotlib* yang digunakan dalam menggambarkan visualisasi, *numpy* yang memiliki *array* yang sangat cepat dan mampu melakukan komputasi besar dalam waktu singkat, *LabelEncoder* yang digunakan untuk mengkodekan data kategorikal menjadi nilai numerik, *RandomForestClassifier* digunakan untuk membuat prediksi kategori, *train_test_split* yang digunakan untuk membagi kumpulan data menjadi subset penelitian dan pengujian, dan *accuracy_score* yang digunakan untuk mencari akurasi model pada dataset penelitian.

2. Melakukan inisialisasi model *Random Forest* dengan menentukan parameter seperti jumlah pohon (*estimators*), kedalaman maksimum pohon, dan lainnya. Pada penelitian ini, parameter yang digunakan ialah jumlah banyak nya pohon (*estimators*) sebanyak 100 dan *atribut random_state* dengan angka acak sebanyak 42.

3. Latih model tersebut dengan menggunakan set pelatihan dengan fungsi yang digunakan, sebagai berikut:

```
# Latih model
clf.fit(x_train, y_train)
```

Gambar 15. Source Code Latih Model

b. Prediksi Model

Setelah membuat model, selanjutnya melakukan prediksi. Gunakan model yang telah dilatih tersebut untuk membuat aplikasi pada set pengujian atau data baru. berikut *source code* yang digunakan:

```
# Buat model Random Forest
clf = RandomForestClassifier(n_estimators=100, random_state=42)
```

Gambar 16. Source Code Membuat Model

c. Evaluasi model

Langkah selanjutnya yaitu melakukan evaluasi kinerja model menggunakan metrik yang relevan, seperti akurasi, persisi, *recall* *F1-score* atau metrik kebingungan (*confusion matrix*). Pada penelitian ini, evaluasi kinerja model yang digunakan ialah akurasi (*accuracy*). Berikut *source code* untuk melakukan evaluasi kinerja model dan hasil dari evaluasi kinerja model:

```
# Evaluasi model
y_pred = clf.predict(x_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

Gambar 17. Source Code Evaluasi Kinerja Model

```
Length of features: 1565
Numerical labels: 1565
Accuracy: 1.0
```

Gambar 18. Hasil dari Evaluasi Kinerja Model

2. Hasil Proses Klasifikasi

Hasil akhir yang diperoleh dari 1.417 sampel benih jagung ialah 695 sampel benih jagung berlabel *Good* dan 722 sampel benih jagung berlabel *Bad*. Tabel berikut menampilkan beberapa sampel yang berhasil diklasifikasikan.

Tabel 3. Beberapa Sampel yang Berhasil Diklasifikasikan

File Name	Contrast	Correlation	Energy	Homogeneity	Area	Perimeter	Label
images/bad\Hasil_	370,9550	0,98716180	0,7415		1476	1967,1038	
ROI_Bad (1).jpg	494	4	96752	0,994295281	38,5	02	<i>bad</i>
images/bad\Hasil_	850,3512	0,96672741	0,7707		1061	1701,5403	
ROI_Bad (10).jpg	397	8	51898	0,986922904	56,5	15	<i>bad</i>
images/bad\Hasil_	355,8424	0,98322451	0,8175		1787	1626,0600	
ROI_Bad (100).jpg	022	8	22934	0,99452769	28,5	86	<i>bad</i>
images/bad\Hasil_	318,4337	0,98581473	0,8061		1867	1669,8590	
ROI_Bad (101).jpg	904	7	65336	0,995102977	30,5	74	<i>bad</i>
images/bad\Hasil_	314,2128	0,98649821	0,7983		1719	1822,6702	
ROI_Bad (102).jpg	251	7	09976	0,995167889	57	56	<i>bad</i>
images/bad\Hasil_	396,0425	0,98176983	0,8123		1667	1592,3616	
ROI_Bad (104).jpg	205	4	12025	0,993909474	65	01	<i>bad</i>
images/bad\Hasil_	273,6035		0,7929		1801	1635,2976	
ROI_Bad (105).jpg	908	0,98853487	77358	0,995792397	72	73	<i>bad</i>

images/bad\Hasil_	289,3351	0,98856669	0,7787		1876	1704,1849	
ROI_Bad (106).jpg	637	1	11529	0,99555047	50	77	<i>bad</i>
images/bad\Hasil_	289,3351	0,98856669	0,7787		1876	1704,1849	
ROI_Bad (108).jpg	637	1	11529	0,99555047	50	77	<i>bad</i>
images/bad\Hasil_	315,9728	0,98533320	0,8147		1706	1857,1139	
ROI_Bad (109).jpg	888	8	72679	0,995140822	21,5	07	<i>bad</i>
images/good\Hasil_	293,2481	0,98740391	0,7984		1878	1839,9667	
_ROI_Good (1).jpg	351	4	23869	0,995490294	40,5	21	<i>good</i>
images/good\Hasil_							
_ROI_Good	264,5293	0,98855335	0,8003		1922	1701,0134	
(10).jpg	773	2	43816	0,995931944	63	03	<i>good</i>
images/good\Hasil_							
_ROI_Good	270,9428	0,98874784	0,7909		1894	1747,8418	
(100).jpg	759	7	13238	0,995833315	37	29	<i>good</i>
images/good\Hasil_							
_ROI_Good	271,8054	0,98776248	0,8088		1905	1718,2438	
(101).jpg	352	7	65629	0,99582005	24	49	<i>good</i>
images/good\Hasil_							
_ROI_Good	269,4376	0,98813338	0,8041		1868	1719,5403	
(102).jpg	394	6	71726	0,995856463	11,5	14	<i>good</i>
images/good\Hasil_							
_ROI_Good	370,9965	0,98465768	0,7889		1818	2037,8073	
(103).jpg	965	1	55483	0,994294642	16	41	<i>good</i>
images/good\Hasil_							
_ROI_Good	273,0759	0,98801845	0,8033		1704	1603,7585	
(104).jpg	581	9	15711	0,995800511	16	7	<i>good</i>
images/good\Hasil_							
_ROI_Good	272,8534	0,98778808	0,8075		2023	1730,2438	
(105).jpg	656	6	96311	0,995803933	77	51	<i>good</i>
images/good\Hasil_							
_ROI_Good	283,3961	0,98680955	0,8156		1759	1617,1311	
(106).jpg	542	3	28625	0,995641802	49	55	<i>good</i>

KESIMPULAN DAN REKOMENDASI

Setelah melakukan penelitian Klasifikasi Kualitas Varietas Benih Jagung Bima 20 Menggunakan Metode *Random Forest* secara keseluruhan menghasilkan bahwa ekstraksi fitur tekstur menggunakan *Gray Level Co-occurrence Matrix* (GCLM) dan ekstraksi fitur bentuk memperoleh nilai yang dapat diklasifikasikan menggunakan metode *random forest*. Hasil klasifikasi yang diperoleh tersebut memiliki tingkat akurasi 100% akurat sesuai dengan pernyataan melalui survei yang dilakukan kepada seorang kepala SMK di Pesantren Teknologi Riau dan juga seorang guru dalam bidang pertanian di Pesantren Teknologi Riau yaitu ibu Azrida Syamsi M.Si. Maka dari itu ekstraksi dan metode yang diterapkan pada penelitian ini berhasil diimplementasikan.

Untuk penelitian selanjutnya, penulis berharap bisa menambahkan parameter lain seperti warna dan deteksi jamur pada benih jagung. Karena dua parameter tersebut mempengaruhi kualitas dari benih jagung. Selain itu, Penelitian selanjutnya juga diharapkan dapat melakukan proses klasifikasi menggunakan parameter warna dan deteksi jamur dengan metode selain *random forest* untuk mengetahui seberapa besar tingkat akurasi dari penelitian tersebut

REFERENSI

- Ali, A., Qadri, S., Mashwani, W. K., Brahim Belhaouari, S., Naeem, S., Rafique, S., Jamal, F., Chesneau, C., & Anam, S. (2020). Machine learning approach for the classification of corn seed using hybrid features. *International Journal of Food Properties*, 23(1), 1097–1111. <https://doi.org/10.1080/10942912.2020.1778724>
- Al Rivan, M. E., Rachmat, N., & Ayustin, M. R. (2020). Klasifikasi Jenis Kacang- Kacangan Berdasarkan Tekstur Menggunakan Jaringan Syaraf Tiruan. *Jurnal Komputer Terapan*, 6(1), 89-98.
- Alwi, L., Hermawan, A. T., & Kristian, Y. (2020). Identifikasi Biji-Bijian Berdasarkan Ekstraksi Random Forest. *Journal of Intelligent Systems and Computation*, 1, 92–98. <https://jurnal.stts.edu/index.php/INSYST/article/view/93>

- Ardiansyah, J., Purnamasari, R., & Bambang, H. (2020). *Dengan Metode Discret Wavelet Transform Dan Klasifikasi Support Vector Machine Berbasis Pengolahan Citra Digital Classification System of Corn Kernel Quality Based on Texture Using Discrete Wavelet Transform Methode and Support Vector Machine Classification*. 511–520.
- Astiningrum, M., Syulistyo, A. R., & Zakariya, M. A. (2020). Warna dan Tekstur Menggunakan HSV Dan Gray Level Run Length Matrix (GLRLM). *JIP (Jurnal Informatika Polinema)*, 7(1), 37–44.
- Chandra, V., Jaman, J. H., Informatika, T., Ilmu, F., Universitas, K., Karawang, S., Neighbor, K., Digital, P. C., & Ciri, E. (2022). Data Citra Digital Menggunakan Algoritma K-Nearest Neighbor. *Jurnal Nuansa Informatika*, 16.
- Effendi, M., Jannah, M., & Effendi, U. (2019, February). *Corn quality identification using image processing with k-nearest neighbor classifier based on color and texture features*. In IOP Conference Series: Earth and Environmental Science (Vol. 230, No. 1, p. 012066). IOP Publishing.
- Jannah, M. (2018). Identifikasi Mutu Jagung Menggunakan Fitur Warna Dan Tekstur Berbasis Pengolahan Citra Digital Dan Algoritma K- Nearest Neighbor (K-NN). 12–26.
- Kishore, B., Yasar, A., Taspinar, Y. S., Kursun, R., Cinar, I., Shankar, V. G., ... & Ofori, I. (2022). Computer-aided multiclass classification of corn from corn images integrating deep feature extraction. *Computational Intelligence and Neuroscience*, 2022.
- Li, X., Dai, B., Sun, H., & Li, W. (2019). Corn classification system based on computer vision. *Symmetry*, 11(4), 591.
- Malang, R. A. A. P. N., & Malang, O. D. T. P. N. Penentuan Klasifikasi Mutu Fisik Beras Dari Bentuk Fisik Dan Warna Menggunakan Metode Connected Component Labelling.
- Muddin, M. I. U., Soedibyo, D. W., & Wahyuningsih, S. (2019). Identifikasi varietas benih jagung (*zea mays l.*) menggunakan pengolahan citra digital berbasis jaringan syaraf tiruan. *Teknika*, 8(2), 78-85.

- Putri, D. A., Munawar, A. A., & Nasution, I. S. (2022). Klasifikasi Mutu Fisik Biji Kopi Beras Robusta menggunakan Pengolahan Citra Digital. *Jurnal Ilmiah Mahasiswa Pertanian*, 7(2).
- Qiu, G., Lü, E., Wang, N., Lu, H., Wang, F., & Zeng, F. (2019). *Cultivar classification of single sweet corn seed using fourier transform near-infrared spectroscopy combined with discriminant analysis*. *Applied Sciences*, 9(8), 1530
- Rosiani, U. D., Mentari, M., & Prastya, A. N. P. (2019). Klasifikasi Kualitas Biji Jagung Berdasarkan Deteksi Warna dan Bentuk Menggunakan Metode K-Nearest Neighbor. In *Seminar Informatika Aplikatif Polinema* (pp. 413-417).
- Septiana, E. R., Fiolana, F. A., & Erwanto, D. (2022). Klasifikasi Kualitas Citra Kedelai Hitam (Malika) Menggunakan Metode K-Nearest Neighbor. *Journal of Electrical Engineering and Computer (JEECOM)*, 4(2), 79-86.