

PROTOTYPE SISTEM SORTING PACK SEMEN DENGAN METODE NEURAL NETWORK (NN)

Afif Mussabiq¹⁾, Denny Irawan²⁾, Rini Puji Astutik³⁾

^{1,2,3)} Jurusan Teknik Elektro, Fakultas Teknik, Universitas Muhammadiyah Gresik
Jl. Sumatra No 101, Gresik 61121, Jawa Timur, Indonesia

E-mail: ¹⁾afifmussabiq23@gmail.com, ²⁾den2mas@umg.ac.id, ³⁾astutik_rpa@umg.ac.id

ABSTRAK

Proses *packing* semen pada PT. Cemindo Gemilang Plant Gresik masih bergantung pada manusia atau petugas yang mengoperasikan sehingga kualitas proses distribusi yang bertaraf internasional menjadi masalah penting terkait *error* pada saat pengecekan kemasan semen. *Object Recognition*, digunakan untuk memperbaiki sistem pengecekan kualitas kemasan semen tersebut dimana untuk citra digital, identifikasi objek membutuhkan teknik dan metode yang mampu mengidentifikasi dan mengklasifikasi fitur dimana komponen utamanya adalah warna sebagai representasi pada citra digital. Penelitian ini bertujuan membuat inovasi desain sistem klasifikasi citra digital sekaligus pemantauan mengenai kualitas kemasan semen dengan ekstraksi fitur menggunakan *Gray Level Co-occurrent Matrix* (GLCM), fitur yang dihasilkan setelah pengolahan GLCM adalah *Correlation, Contrast, Energy dan Homogeneity*. Fitur tersebut kemudian diklasifikasi menggunakan *Neural Network* (NN) untuk menentukan kualitas kemasan. Setelah itu ditransmisikan fungsinya pada mikrokontroler Arduino dan selanjutnya dilakukan tindakan yang tepat guna proses sortir untuk mendeteksi kualitas kemasan yang layak ataupun tak layak masuk dalam proses selanjutnya. Semua informasi ditampilkan dalam aplikasi *Visual GUI Matlab*. Hasil dari ekstraksi fitur menggunakan GLCM bekerja dengan akurat dalam klasifikasi kualitas kemasan menggunakan NN. Pengujian dilakukan dengan menggunakan *prototype* sistem *sortir pack* dan dimonitoring menggunakan GUI Matlab. Tingkat keberhasilan mencapai 75% dari delapan kali pengujian.

Kata kunci: Kamera, Arduino Uno, *Neural Network* (NN), *Gray Level Cooccurrence Matrix* (GLCM)

ABSTRACT

Cement packing process at PT. Cemindo Gemilang Plant Gresik still depends on humans or officers who operate it, so the quality of the international standard distribution process is an important problem related to errors when checking cement packaging. Object Recognition is used to improve the cement packaging quality checking system, where for digital images, object identification requires techniques and methods that are able to identify and classify features where the main component is color as a representation in the digital image. This research aims to innovate the design of a digital image classification system as well as monitor the quality of cement packaging by extracting features using the Gray Level Co-occurrent Matrix (GLCM). The features produced after GLCM processing are Correlation, Contrast, Energy and Homogeneity. These features are then classified using a Neural Network (NN) to determine packaging quality. After that, the function is transmitted to the Arduino microcontroller and then appropriate actions are taken for the sorting process to detect the quality of the packaging which is suitable or not suitable for entering the next process. All information is displayed in the Visual GUI Matlab application. The results of feature extraction using GLCM work accurately in packaging quality classification using NN. Testing was carried out using a prototype pack sorting system and monitored using the Matlab GUI. The success rate reached 75% from eight tests.

Keywords: Camera, Arduino Uno, *Neural Network* (NN), *Gray Level Cooccurrence Matrix* (GLCM)

1. PENDAHULUAN

PT. Cemindo Gemilang merupakan salah satu perusahaan milik swasta yang bergerak di bidang semen. Selama ini produk olahan dari PT. Cemindo Gemilang sangatlah dipercaya oleh masyarakat. Produk dari PT. Cemindo Gemilang yaitu Semen yang mempunyai merk dagang Semen Merah Putih. Namun pada penelitian ini akan dibahas mengenai perbaikan *packaging* semen. Karena tingginya konsumen dan permintaan, pihak produksi industri semen mencoba untuk memperbaiki kualitas produk semen salah satunya adalah memilah (*sorting*) hasil *packaging* semen yang cacat.

Pengambilan produk cacat yang ada di salah satu perusahaan pembuatan semen di Jawa Timur masih dengan cara manual yang masih banyak membutuhkan operator untuk menyelesaikan masalah tersebut, sehingga sehubungan dengan hal itu, dalam proses pengambilan produk cacat merupakan salah satu sistem yang perlu diperhatikan karena berpengaruh dengan hasil produksi. Penelitian ini membahas secara keseluruhan tentang sistem otomatis dalam suatu proses didalam pabrik yaitu *Rejector*.

Rejector adalah suatu sistem pemilah (*Sorting*) atau proses pemisahan produk dengan memberikan indikator atau penggolongan produk tertentu [1]. Oleh sebab itu untuk membedakan produk sempurna dengan produk cacat dan efisiensi waktu produksi, maka tujuan penelitian ini adalah men-desain suatu sistem yaitu “Prototype Sistem Sorting Pack Semen Dengan Metode Neural Network (NN)”.

Berkaitan dengan permasalahan diatas, maka penelitian ini diharapkan dapat melakukan proses klasifikasi sekaligus pemantauan mengenai kualitas sistem sortir menggunakan NN [2].

Ekstraksi fitur digunakan untuk pengambilan informasi dari citra kemasan semen, fitur yang dihasilkan setelah pengolahan yaitu kontras, korelasi, energi dan homogenitas. Pada penelitian ini hasil ekstraksi fitur akan diklasifikasi menggunakan NN untuk menentukan kualitas kemasan semen, kemudian ditransmisikan fungsinya pada mikrokontroler dan selanjutnya dilakukan tindakan yang tepat

guna proses sortir untuk mendeteksi kualitas kemasan semen yang layak ataupun tak layak masuk dalam proses selanjutnya. Semua informasi ditampilkan dalam aplikasi Visual GUI Matlab [3].

2. METODE PENELITIAN

1. Perencanaan/*Planning*

Perencanaan atau *Planning* adalah langkah awal dalam melakukan penelitian. Langkah ini menjadi landasan bagi untuk berikutnya, yaitu pelaksanaan, observasi dan refleksi. Meskipun, pelaksanaan tindakan memiliki nilai strategis dalam kegiatan penelitian, namun tindakan tersebut tidaklah berdiri sendiri, melainkan merupakan bagian yang tidak terpisahkan dari kegiatan perencanaan.

2. Analisis

Berikutnya adalah melakukan analisis, yang proses lengkapnya adalah sebagai berikut : pengumpulan data, penyusunan dan analisa data yang dibutuhkan untuk desain sistem. Proses analisis data itu dimulai dari menelaah data secara keseluruhan yang telah tersedia dari berbagai macam sumber, baik itu pengamatan, wawancara, catatan lapangan dan yang lainnya. Data ini dapat ditemukan dengan cepat. Dalam penelitian ini yang menjadi sumber data sekunder adalah literatur, artikel, jurnal serta situs di internet yang berkenaan dengan penelitian yang dilakukan.

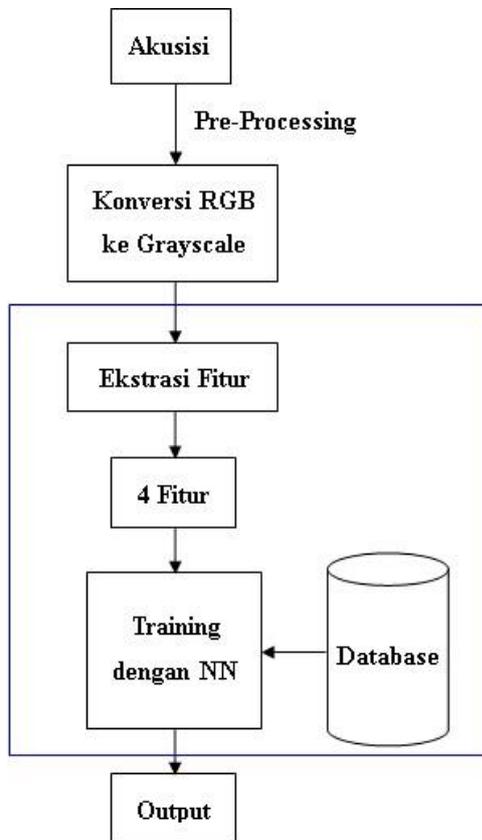
3. Rancangan atau Desain

Rancangan penelitian adalah suatu cara yang akan digunakan dalam pelaksanaan penelitian dan menjelaskan setiap prosedur penelitian mulai dari tujuan penelitian sampai dengan analisis data. Komponen yang umumnya teradap dalam rancangan penelitian adalah : tujuan penelitian, jenis penelitian yang digunakan dan teknik pengumpulan data.

a. Perancangan sistem

Pada gambar 2.1 dijelaskan ada 3 tahap, yaitu *Pre-Processing*, *Processing*, dan *Classifying* [4]. Pada tahap *Pre-Processing*, input citra akan diolah dan diubah dari RGB ke *Grayscale*. Tahapan berikutnya adalah *Processing*. Pada tahap ini, input citra akan masuk tahap awal,

yaitu proses konvolusi. Proses konvolusi akan membuat *Array* baru atau input baru.

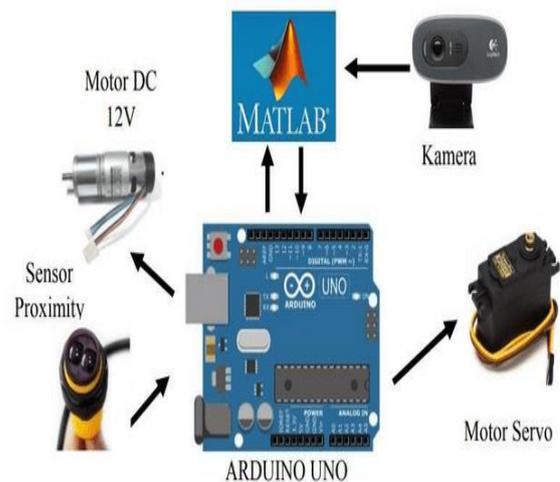


Gambar 2.1 Arsitektur sistem

Setelah itu, selanjutnya adalah proses *Maxpooling* [5], proses ini berfungsi untuk mempercepat proses komputasi pada program dan juga menghilangkan *noise* pada input citra yang telah diambil. Setelah itu, masuk ke proses inti yaitu proses *Training dengan Neural Network* (NN) [6]. Pada proses ini akan menghasilkan bobot yang akan digunakan untuk klasifikasi output [7], [8].

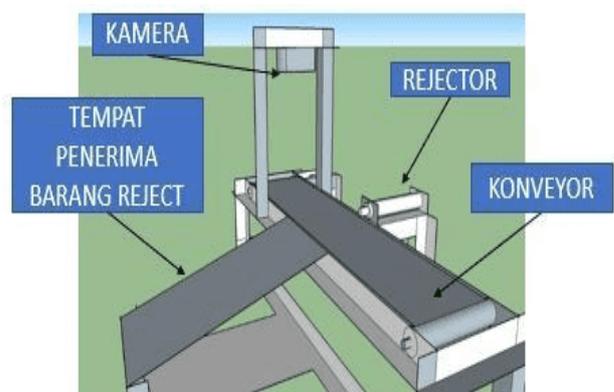
- b. Perancangan desain *hardware*
Hardware yang akan digunakan pada sistem *sorting* kemasan semen seperti disajikan pada gambar 2.2. Dari gambar tersebut dapat diketahui untuk sensor, aktuator dan kontroler yang akan digunakan. Pada aktuator terdapat motor DC yang akan berfungsi sebagai penggerak konveyor. Untuk sensornya

digunakan untuk mendeteksi adanya objek yang akan dideteksi. Kemudian, pada kontroler terdapat PC [9] yang berfungsi sebagai kontroler yang mengirim data hasil kamera ke Arduino [10] dan menerima data dari sensor dan kamera [11].



Gambar 2.2 Perancangan desain *hardware*

- c. Perancangan desain mekanik
 Perancangan desain mekanik (yang dibedakan dengan desain *hardware*) seperti pada gambar 2.3. *Sorting* kemasan semen akan berjalan otomatis melalui konveyor. Ketika sensor mendeteksi adanya objek maka kamera akan aktif dan konveyor akan berhenti. Setelah itu proses algoritma berjalan. Sistem ini akan memilih kemasan semen dengan klasifikasi yang telah ditentukan.



Gambar 2.3 Desain 3D rancang bangun sistem *sorting* kemasan semen

4. Implementasi

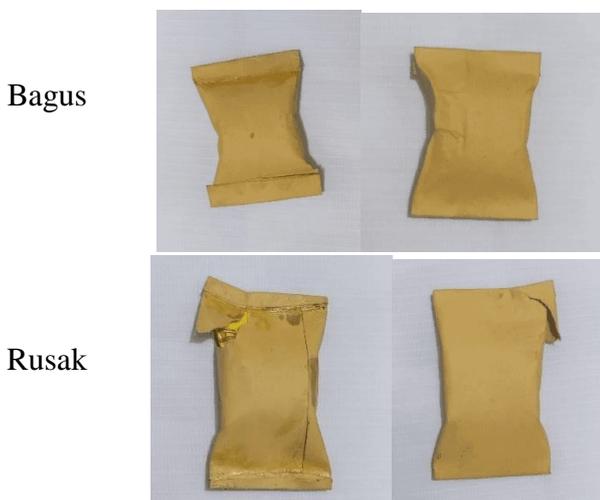
Implementasi dapat dimaksudkan sebagai suatu aktivitas yang berkaitan dengan penyelesaian suatu pekerjaan dengan penggunaan sarana (alat) dengan acuan dari aturan yang berlaku untuk memperoleh hasil.

3. HASIL DAN DISKUSI

3.1 Pembentukan Datasheet

Berdasarkan berbagai referensi penelitian seputar aplikasi ekstraksi fitur dan jaringan syaraf tiruan, langkah awal sebelum proses training dimulai adalah pembagian dataset untuk keperluan training. Pada penelitian ini proporsi data training yang digunakan adalah 80% dari total data dan untuk keperluan pengujian atau validasi sebanyak 20%. Total data secara keseluruhan adalah 90 data dengan rincian data citra untuk kelas good sebanyak 30 data, data citra untuk kelas reject sebanyak 30 data. Semua citra berukuran sama yaitu 320 x 240 piksel. Penentuan proporsi data untuk keperluan training dan validasi dilakukan secara acak atau random.

Hasil dibawah ini adalah sampel dataset untuk masing – masing kelas.

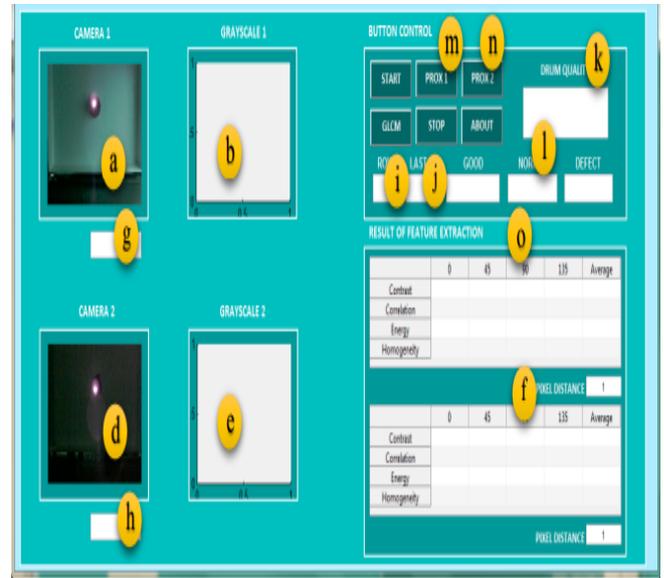


Gambar 3.1 Kondisi kantong semen bagus dan rusak

Dari semua total dataset yang digunakan ada 72 (80%) data citra digunakan untuk keperluan training dan 18 (20%) data citra untuk keperluan validasi.

3.2 Tampilan Interface

Pada gambar 3.2 merupakan desain interface dari GUI Matlab untuk prototype system sortir.



Gambar 3.2 Tampilan interface pada GUI Matlab

Keterangan :

- a : *Display* citra RGB dari kamera 1 (*realtime*)
- b : *Display* konversi RGB ke *Grayscale* (kamera 1)
- c : Digunakan untuk *close* GUI Matlab
- d : *Display* citra RGB dari kamera 2 (*realtime*)
- e : *Display* konversi RGB ke *Grayscale* (kamera 1)
- f : *Display* nilai ekstraksi fitur dengan berbagai sudut pada kamera 2
- g : Hasil 25 klasifikasi citra dari kamera 1
- h : Hasil klasifikasi citra dari kamera 2
- i : Digunakan untuk pengambilan data dari GUI ke Ms.Excel
- j : Digunakan untuk penanda pengambilan data terakhir dari baris dan kolom Ms.Excel
- k : Hasil klasifikasi fungsi AND dari hasil klasifikasi citra kamera 1 dan 2
- l : Memiliki 2 fungsi : sebagai set output saat pengambilan data dan output saat *running realtime*
- m : Digunakan sebagai *button callback* GUI untuk *capture* pada kamera 1
- n : Digunakan sebagai *button callback* GUI untuk *capture* pada kamera 2

o : Digunakan sebagai *display* nilai ekstraksi fitur dengan berbagai sudut pada kamera 1

3.3 Ekstraksi Fitur GLCM

Proses ekstraksi fitur dilakukan dengan konversi gambar RGB berukuran 320x240 *pixel* ke dalam *Grayscale*. Pada penelitian ini setelah mendapatkan hasil dari *pre-processing* kemudian dilanjutkan dengan pembuatan matriks *co-occurrence* dengan mengisi jumlah hubungan spasial yang ada pada matriks *Grayscale*.

a) Berikut ini merupakan perhitungan manual GLCM dengan jarak *pixel* $d = 1$ *pixel* dan sudut $\theta = 0^\circ$.

1. Matriks I yaitu merupakan matriks *Grayscale* berukuran 4x4 yang akan diproses ke dalam kuantisasi dengan 8 gradasi.

$$\text{Matriks } \textit{Greyscale} = \begin{bmatrix} 20 & 0 & 55 & 0 \\ 39 & 35 & 0 & 17 \\ 80 & 28 & 0 & 36 \\ 120 & 10 & 0 & 93 \end{bmatrix}$$

$$\begin{aligned} & (20/256) + (0/256) + (55/256) + (0/256) + (39/256) + (35/256) + \\ & (0/256) + (17/256) + (80/256) + (28/256) + (0/256) + (36/256) + \\ & (120/256) + (10/256) + (0/256) + (93/256) \\ & = (0.078) + (0) + (0.215) + (0) + (0.152) + (0.137) + (0) + (0.066) + \\ & (0.313) + (0.109) + (0) + (0.141) + (0.469) + (0.039) + (0) + (0.363) \\ & = 2.082 \end{aligned}$$

2. Pada tahap kuantisasi 8, gradasi akan dilakukan teknik digitalisasi dalam pengolahan teknik citra digital yang bertujuan untuk mendapatkan nilai intensitas atau kecerahan *pixel* dengan cara mengelompokkan nilai intensitas *pixel* ke dalam beberapa level. Gambar 3.1 diperoleh dengan memperhatikan tingkat keabuan hasil matriks kookuren sebuah citra dengan citra aslinya menggunakan Matlab. Pada proses ini menghasilkan matriks kuantisasi yang kemudian akan dilanjutkan dengan mencatat hubungan ketetangaan antar *pixel* dalam bentuk matriks kookurensi.

Tabel 3.1 Derajat keabuan 8 level

Level	Nilai
0	0 – 31
1	32 – 63
2	64 – 95
3	96 – 127
4	128 – 159
5	160 – 191
6	192 – 233
7	224 – 256

$$\text{Matriks Kookuren} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 2 & 0 & 0 & 1 \\ 3 & 0 & 0 & 2 \end{bmatrix}$$

$$\begin{aligned} & (0/8) + (0/8) + (1/8) + (0/8) + (1/8) + (1/8) + (0/8) + (0/8) + (2/8) + \\ & (0/8) + (0/8) + (1/8) + (3/8) + (0/8) + (0/8) + (2/8) \\ & = (0) + (0) + (0.125) + (0) + (0.125) + (0.125) + (0) + (0) + (0.250) + \\ & (0) + (0) + (0.125) + (0.375) + (0) + (0) + (0.250) \\ & = 1.375 \end{aligned}$$

3. Proses berikutnya merupakan tahap pembentukan matriks kookurensi dari matrix I. Hubungan ketetangaan antara *pixel* 0 dengan 0, 0 dengan 1 dan *pixel* 1 dengan 0 dst. Setelah semua elemen matriks dicari hubungan ketetangaan, langkah selanjutnya adalah dilakukan proses penjumlahan matriks kookuren dengan transpose matrix agar membentuk simetris antara sudut 0° dan sudut 180° .

Transpose Matriks Kookuren =

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 2 & 0 & 0 & 1 \\ 3 & 0 & 0 & 2 \end{bmatrix}$$

	0	1	2	3
0	4	2	1	0
1	2	1	0	0
2	1	0	0	0
3	1	0	0	0

4. Transpose matriks kookurensi digunakan untuk mendapatkan sudut simetrisnya yaitu sudut 180° .

$$I \text{ kookurensi} = \begin{bmatrix} 4 & 2 & 1 & 0 \\ 2 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$I^T = \begin{bmatrix} 4 & 2 & 1 & 1 \\ 2 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

5. Operasi matriks I kookurensi + I^T untuk mendapatkan matrix I simetris antara sudut 0° dan sudut 180°. Dan langkah selanjutnya adalah pembentukan matriks kookurensi.

$$\begin{bmatrix} 4 & 2 & 1 & 0 \\ 2 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 4 & 2 & 1 & 1 \\ 2 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 8 & 4 & 2 & 1 \\ 4 & 2 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

I kookurensi I^T I simetris

6. Mengubah matriks I simetris menjadi I simetris ternormalisasi dengan persamaan 1.

I Simetris Ternormalisasi =

$$\frac{I \text{ simetris}}{\sum n \text{ pixel}} \quad (1)$$

$$\begin{aligned} & (8/22) + (4/22) + (2/22) + (1/22) + (4/22) + (2/22) + (0/22) + (0/22) + \\ & (2/22) + (0/22) + (0/22) + (0/22) + (1/22) + (0/22) + (0/22) + (0/22) \\ & = (0.363) + (0.181) + (0.090) + (0.045) + (0.181) + (0.090) + (0.090) + (0.045) \\ & = 1.085 \end{aligned}$$

Tahap normalisasi ini dengan cara membagi masing – masing elemen matriks dengan jumlah *pixel*.

$$\begin{bmatrix} 8 & 4 & 2 & 1 \\ 4 & 2 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

I simetris

$$\begin{bmatrix} 8/22 & 4/22 & 2/22 & 1/22 \\ 4/22 & 2/22 & 0 & 0 \\ 2/22 & 0 & 0 & 0 \\ 1/22 & 0 & 0 & 0 \end{bmatrix}$$

Proses normalisasi

$$\begin{bmatrix} 0.363 & 0.181 & 0.090 & 0.045 \\ 0.181 & 0.090 & 0 & 0 \\ 0.090 & 0 & 0 & 0 \\ 0.045 & 0 & 0 & 0 \end{bmatrix}$$

I ternormalisasi

b) Setelah didapatkan nilai matrix I ternormalisasi, proses selanjutnya adalah menghitung nilai ciri statistiknya. Dibawah ini merupakan fitur vector dari matrix I ternormalisasi.

$$\begin{matrix} \xrightarrow{j} \\ \begin{bmatrix} 0.363 & 0.181 & 0.090 & 0.045 \\ 0.181 & 0.090 & 0 & 0 \\ 0.090 & 0 & 0 & 0 \\ 0.045 & 0 & 0 & 0 \end{bmatrix} \\ \uparrow i \end{matrix}$$

Fitur vector

Dibawah ini merupakan persamaan tahap intepretasi citra :

1. Contrast

$$\text{Contrast} = \sum_i^n \sum_j^m (i-j)^2 c(i,j) \quad (2)$$

$$\begin{aligned} & (0)^2 (0.363) + (0)^2 (0.181) + (0)^2 (0.090) + (0)^2 (0.045) + (0)^2 (0.181) + \\ & (0)^2 (0.090) + (0)^2 (0.090) + (0)^2 (0.045) = 0 \end{aligned}$$

Pada fitur contrast i dan j adalah sifat keabuan dari resolusi 2 *pixel* yang berdekatan. Syaratnya adalah ketika nilai i dan j sama, sel berada pada diagonal dan (i-j) = 0. Nilai – nilai ini merepresentasikan *pixel* yang keseluruhannya mirip dengan tetangga mereka, sehingga diberi bobot 0. Dibawah ini merupakan contoh perhitungan dari fitur contrast pada persamaan 2.

2. Correlation

Correlation =

$$\sum_i^n \sum_j^m \frac{(i - \mu_i)(j - \mu_j) \cdot C(i, j)}{\sqrt{\sigma_i^2 \sigma_j^2}} \quad (3)$$

$$(0.363(0 - 0)^2) + (0.181(0 - 1)^2) + (0.090(0 - 2)^2) + (0.045(0 - 3)^2) + (0.181(1 - 0)^2) + (0.090(1 - 1)^2) + (0.090(2 - 0)^2) + (0.045(3 - 0)^2) = 2.345$$

Mengukur linearitas dari sejumlah pasangan pixel (pairs). Sehingga dapat memberikan petunjuk adanya ukuran ketergantungan struktur linear dalam citra. Dibawah ini merupakan contoh perhitungan dari persamaan 3.

Perhitungan rata-rata nilai kolom dengan i pada persamaan 4.

$$\mu_i = \sum_i^n \sum_j^n (i) (C(i, j)) \quad (4)$$

$$(0)(0.363) + (1)(0.181) + (2)(0.090) + (3)(0.045) + (0)(0.181) + (1)(0.090) + (0)(0.090) + (0)(0.045) = 0.5860$$

Perhitungan rata-rata nilai baris dengan j pada persamaan 5.

$$\mu_j = \sum_i^n \sum_j^n (j) (C(i, j)) \quad (5)$$

$$(0)(0.363) + (0)(0.181) + (0)(0.090) + (0)(0.045) + (1)(0.181) + (1)(0.090) + (2)(0.090) + (3)(0.045) = 0.5860$$

Perhitungan variansi pada matrix i ditunjukkan pada persamaan 6.

$$\sigma_i^2 = \sum_i^n \sum_j^n ((C(i, j))(i - \mu_i)^2) \quad (6)$$

$$((0.363) (0 - 0.5860)^2) + ((0.181) (1 - 0.5860)^2) + ((0.090) (2 - 0.5860)^2) + ((0.045) (3 - 0.5860)^2) + ((0.181) (0 - 0.5860)^2) + ((0.090) (1 - 0.5860)^2) + ((0.090) (0 - 0.5860)^2) + (0.045) ((0 - 0.5860)^2) = 0.7218$$

Perhitungan variansi pada matrix j ditunjukkan pada persamaan 7.

$$\sigma_j^2 = \sum_i^n \sum_j^n ((C(i, j))(j - \mu_j)^2) \quad (7)$$

$$((0.363) (0 - 0.5860)^2) + ((0.181) (0 - 0.5860)^2) + ((0.090) (0 - 0.5860)^2) + ((0.045) (0 - 0.5860)^2) + ((0.181) (1 - 0.5860)^2) + ((0.090) (1 - 0.5860)^2) + ((0.090) (2 - 0.5860)^2) + (0.045) ((3 - 0.5860)^2) = 0.7218$$

3. Energy

$$Energy = \sum_i^n \sum_j^m C^2(i, j) \quad (8)$$

$$(0.363)^2 + (0.181)^2 + (0.090)^2 + (0.045)^2 + (0.181)^2 + (0.090)^2 + (0.090)^2 + (0.045)^2 = 0.228715$$

Energy digunakan untuk menentukan intensitas keabuan dengan ukuran konsentrasi pasangan tertentu pada matriks.

4. Homogeneity

Homogeneity =

$$\sum_i^n \sum_j^m \frac{C(i, j)}{1 + |i + j|} \quad (9)$$

$$\frac{0.363}{1+(0-0)^2} + \frac{0.181}{1+(0-1)^2} + \frac{0.090}{1+(0-2)^2} + \frac{0.045}{1+(0-1)^2} + \frac{0.181}{1+(0-0)^2} + \frac{0.090}{1+(0-1)^2} + \frac{0.090}{1+(0-0)^2} + \frac{0.045}{1+(0-1)^2} = 0.835$$

Fitur *homogeneity* digunakan untuk menentukan jumlah level keabuan akan semakin tinggi jika seragam. Perhitungan dari fitur contrast ditunjukkan pada persamaan 9.

Matriks kookurensi yang akan dibuat adalah hubungan spasial $d=1$ dan 0° . Selanjutnya matriks *co-occurrence* yang sudah didapatkan, harus di-*transpose* untuk mengubah matriks baris menjadi kolom, tujuan dari proses ini adalah untuk mendapatkan sudut simetrisnya yaitu sudut 180° , kemudian dilanjutkan dengan pembuatan matriks simetris dengan cara menjumlahkan matrix *co-occurrence* dengan matriks *transpose* agar membentuk sudut 0° dan 180° . Selanjutnya matriks simetris harus di-normalisasi nilai-nilainya dengan membaginya dengan jumlah total kemungkinan (jumlah nilai pixel). Hal ini dimaksudkan agar nilai-nilai tersebut dikonversi menjadi probabilitas, dimana jika semua nilainya dijumlah akan sama dengan 1. Setelah mendapatkan matriks simetris ter-normalisasi, harus dilakukan ekstraksi informasi statistiknya. Pada penelitian ini menggunakan *contrast*, *correlation*, *energy* dan *homogeneity* untuk menginterpretasikan tekstur citra. Pada dasarnya setiap fitur memiliki nilai yang berbeda tergantung pada kegunaan masing – masing fitur yang disebutkan pada bab sebelumnya.

3.4 Training dengan 16 Hidden Neuron

Percobaan training kedua mempunyai konfigurasi sebagai berikut :

Jumlah iterasi = 385
 Fungsi aktivasi = sigmoid
 Learning rate = 0.2

Dari 385 *epoch* yang dilakukan, *performance* terbaik jaringan (nilai *error* paling mendekati target) adalah pada angka *error* : 0,01083 pada *epoch* ke 384.

Tabel 3.2 Percobaan Hasil Training dengan Output Defect

No.	Target Pengujian	Vektor Input (Biner)	Prediksi NN (Biner)	Keterangan
1.	Defect	[0 0 1]	[0 0 1]	Benar
2.	Defect	[0 0 1]	[0 0 1]	Benar
3.	Defect	[0 0 1]	[0 0 1]	Benar
4.	Defect	[0 0 1]	[0 0 1]	Benar
5.	Defect	[0 0 1]	[0 0 1]	Benar
6.	Defect	[0 0 1]	[0 0 1]	Benar
Akurasi				100%

Tabel 3.3 Percobaan Hasil Training dengan Output Normal

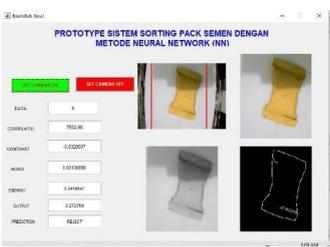
No.	Target Pengujian	Vektor Input (Biner)	Prediksi NN (Biner)	Keterangan
1.	Normal	[0 1 0]	[0 1 0]	Benar
2.	Normal	[0 1 0]	[0 1 0]	Benar
3.	Normal	[0 1 0]	[0 1 0]	Benar
4.	Normal	[0 1 0]	[0 1 0]	Benar
5.	Normal	[0 1 0]	[0 1 0]	Benar
6.	Normal	[0 1 0]	[0 1 0]	Benar
Akurasi				100%

Tabel 3.4 Percobaan Hasil Training dengan Output Good

No.	Target Pengujian	Vektor Input (Biner)	Prediksi NN (Biner)	Keterangan
1.	Good	[1 0 0]	[1 0 0]	Benar
2.	Good	[1 0 0]	[1 0 0]	Benar
3.	Good	[1 0 0]	[1 0 0]	Benar
4.	Good	[1 0 0]	[1 0 0]	Benar
5.	Good	[1 0 0]	[1 0 0]	Benar
6.	Good	[1 0 0]	[1 0 0]	Benar
Akurasi				100%

3.5 Pengujian *Realtime* dengan GUI Matlab

Tabel 3.5 Hasil Klasifikasi Kemasan dengan Ekstraksi Fitur dan NN

No.	Hasil Klasifikasi	Keterangan
1		Benar
2		Salah
3		Benar
4		Benar

Dari Tabel 3.5 dapat dijelaskan bahwa kondisi akan mempengaruhi kualitas kemasan, jika kondisi kemasan rusak/sobek maka terdeteksi "Salah" dan jika kondisi pack tidak sobek/bagus maka terdeteksi "Benar".

Tabel 3.6 Presentase Keberhasilan Pengujian

No.	Kategori Percobaan	Total	Hasil Klasifikasi	
			Salah	Benar
1	Bagus	4	1	3
2	Rusak	4	1	3
Total Data		8	2	6
Presentase Keberhasilan		75%		

Pada Tabel 3.6 dapat disimpulkan bahwa presentase keberhasilan pada saat *realtime* mencapai 75% dari total data keseluruhan adalah 8 dengan rincian percobaan dengan kelas Bagus sebanyak 4 data dan data citra untuk kelas Rusak sebanyak 4 data. Semua citra berukuran 320x240 *pixel*.

4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Penerapan metode NN pada prototype system ini dapat berjalan dengan baik dengan 16 hidden neuron menghasilkan akurasi system sebesar 100%. Namun pada saat pengujian *realtime*, akurasi menurun sebesar 25% sehingga akurasi mencapai 75% saja.

4.2 Saran

Karena pada penelitian ini membahas mengenai ekstraksi fitur maka akan lebih baik apabila kamera yang digunakan memiliki resolusi yang tinggi sehingga lebih detail dalam proses ekstraksi citra.

5. DAFTAR PUSTAKA

- [1] M. F. Assegaf, “*PERANCANGAN ALAT LABEL REJECTOR KEMASAN MENGGUNAKAN RASPBERRY PI DENGAN METODE MACHINE LEARNING*,” Universitas Diponegoro, Yogyakarta, 2021.
- [2] A. M. Zamani, B. Amaliah, and A. Munif, “Implementasi Algoritma Genetika pada Struktur Backpropagation Neural Network untuk Klasifikasi Kanker Payudara,” *Jurnal Teknik ITS*, vol. 1, no. 1, pp. A222–A227, 2012.
- [3] S. Adhi Harmoko, B. Kusumoputro, and M. Rangkuti, “Ekstraksi Ciri Gray Level Co-Occurrence Matrix Dan Probabilistic Neural Network Untuk Pengenalan Cacat Pengelasan,” 2004.
- [4] R. Munir, “Pengolahan citra digital dengan pendekatan algoritmik,” *Informatika, Bandung*, vol. 260, 2004.
- [5] D. Putra, *Pengolahan Citra Digital*. Yogyakarta: ANDI, 2010.
- [6] L. V Fausett, *Fundamentals of neural networks: architectures, algorithms and applications*. Pearson Education India, 2006.
- [7] S. K. Dewi, *Artificial Intelligence (Teknik dan Aplikasi)*. Andi: Yogyakarta, 2003.
- [8] S. Sandi, “Artificial Intelligence,” *Andi Offset*. Andi Offset, Yogyakarta, 1993.
- [9] M. C. Wijaya and A. Prijono, “*Pengolahan Citra Digital Menggunakan Matlab*,” Bandung: Informatika, 2007.
- [10] H. Santoso, *Buku Monster Arduino, Belajar Konsep Dasar Arduino Volume 3*. Jakarta: Elang Sakti, 2017.
- [11] R. Harry and D. Yohannes, “*Teknik Pemrograman Menggunakan Matlab*.” Grasindo: Jakarta, 2007.