

Desain Sistem Pewarnaan Citra Otomatis Berbasis Fitur Tekstur GLCM (*Gray Level Co-Occurrence Matrix*) Menggunakan Backpropagation

Muhammad Sipan¹, Roni Kartika P.², Puri Muliadhi³

¹Program Studi Teknik Elektro, Universitas Semarang

²Program Studi Teknik Elektro, Universitas Semarang

³Program Studi Teknik Elektro, Universitas Semarang

zh1_puan@usm.ac.id, ronikartika@usm.ac.id, puri@usm.ac.id

Abstract— Sebuah desain atau model merupakan cara untuk mempermudah seseorang untuk menterjemahkan sesuatu yang sulit untuk di mengerti, misalkan memodelkan bagaimana cara membedakan sebuah retina mata, memodelkan sebuah proses produksi, memodelkan untuk mendeskripsikan sebuah benda dan lainlain. Penelitian ini membahas bagaimana memodelkan sebuah pewarnaan otomatis citra berbasis fitur tekstur GLCM (*Gray level co-occurrence matrix*) dengan menggunakan Backpropagation. Penelitian ini menjelaskan citra grayscale yang akan di warnai dengan memperhatikan fitur yang ada pada citra tersebut untuk di bandingkan dengan citra warna seberapa persamaan nilai-nilai fitur tekstur GLCM pada kedua citra. Setelah ditemukan persamaan atau kedekatan nilai fitur tekstur GLCM tersebut maka kedua citra tersebut memiliki kedekatan atau kesamaan warna. Kemudian warna dari citra warna tersebut kita transfer ke citra grayscale sehingga dihasilkan citra grayscale yang terwarnai menjadi citra yang berwarna. Penyamaan fitur tekstur dilakukan dengan membandingkan setiap blok kedua citra yaitu citra warna dan citra grayscale dengan cara citra di bagi menjadi blok blok citra, dimana blok-blok citra tersebut kita cari fitur teksturnya. Blok-blok tersebut nantinya akan dibandingkan dengan blok lain pada masing-masing citra sehingga di temukan blok yang memiliki nilai fitur tekstur yang sama atau berdekatan nilainya.

Nilai-nilai fitur tekstur tersebut dijadikan acuan untuk di proses menggunakan metode backpropagasi. Metode ini akan memproses data dimana proses tersebut dibagi dua, yaitu proses training dan proses testing untuk memperoleh hasil citra yang diharapkan, yaitu blok citra warna yang paling mendekati kemiripan atau sama dengan blok citra grayscale untuk kemudian dilanjutkan proses transfer warna ke citra grayscale, sehingga citra tersebut menjadi citra warna

Kata kunci— *Backpropagation, Citra, Desain Sistem, Grayscale, GLCM, Model.*

I. PENDAHULUAN

Sebuah citra dapat dikatakan sebagai visualisasi dari sebuah bentuk, baik beraturan maupun bentuk yang tidak beraturan (Alexandru, Ivanovici, & Richard, 2014). Kita dapat memvisualisasikan dengan baik apabila citra tersebut merupakan citra warna, dengan citra warna kita bisa memvisualisasikan sebuah citra dengan detail dan terinci, misal kita bisa mendeskripsikan sebuah laut, air laut itu biru, mendeskripsikan sebuah bunga melati yaitu bunga melati itu putih dan memiliki kelopak yang kecil-kecil dengan bentuk yang elips lonjong dan lain sebagainya.

Deskripsi ini tidak bisa kita lakukan pada sebuah citra grayscale, jika kita bisa mendeskripsikan citra grayscale adalah hanya berupa bentuk sebuah benda, karena citra grayscale hanya memvisualisasikan sebuah citra dengan hitam dan putih saja. Untuk mewarnai sebuah citra grayscale juga bukan hal yang mudah, karena di perlukan cara yang susah dan langkah yang panjang. Berdasarkan permasalahan tersebut penulis membuat model untuk mewarnai citra berbasis fitur GLCM dengan metode backpropagation agar dapat digunakan sebagai panduan dalam mewarnai sebuah citra.

Penelitian tentang pewarnaan sering dilakukan dari tahun-ketahun baik yang otomatis atau semi otomatis dengan berbagai metode yang digunakan untuk menyelesaikan permasalahan tersebut dimana memerlukan sejumlah interaksi penggunaannya.

(Liu & Zhang, 2012) membahas tentang pewarnaan otomatis menggunakan regresi histogram dengan menggunakan citra warna sebagai citra sumber sebagai patokan citra grayscale (citra target), dengan mencari regresi local masing-masing citra dan menyelaraskan nilai histogramnya untuk mewarnai citra grayscale.

(Alekseev, Rozaliev, & Orlova, 2015) menjelaskan bagaimana pewarnaan citra grayscale otomatis menggunakan analisis *intelligent scene* yang digunakan untuk mentransfer warna dari citra sumber ke citra grayscale yang dipilih.

(Hogervorst & Toet, 2010) membahas tentang pewarnaan citra dan video semi-otomatis, yaitu dengan menandai area yang akan di warnai dengan penanda warna citra sumber, pewarnaan video dengan memberi penanda

dan informasi chrominance pada frame referensi sebagai acuan warna video monokrom.

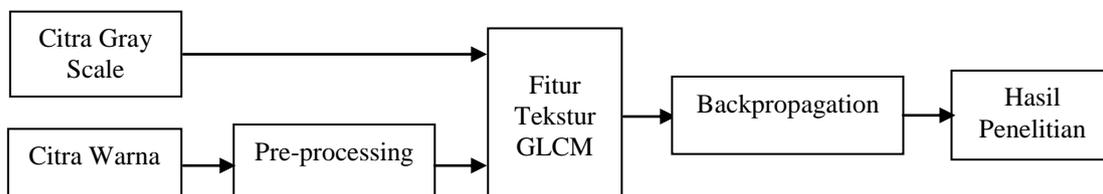
(Zhong Zhen, Gui Yan, Ma Lizhuang, 2010), membahas tentang pewarnaan berbasis tekstur dengan wavelet gabor yang digunakan untuk analisis tekstur dan kontinuitas pola citra, kemudian mensegmentasi tekstur dalam kelas-kelas yang tertentu kemudian kelas-kelas tersebut digunakan acuan untuk mewarnai citra target.

(Kolmogorov & Rother, 2007), membahas tentang pewarnaan berbasis graph cuts, penggunaan graph cuts untuk menandai warna pada tiap area yang berbeda tiap beberapa piksel yang dipilih. Keuntungan utama dari metode ini adalah modifikasi dan pemurnian warna citra tidak menyebabkan recomputation pada proses pewarnaan citra dan dilakukan sangat cepat, berkat Sifat evolusi dari kisi maps yang digabungkan.

(Sousa & Blaes, n.d.), penelitiannya membahas bagaimana mewarnai citra grayscale yang kondisinya rusak menggunakan algoritma coloration, dengan memperbaiki citra grayscale dahulu dengan filter bilateral berbasis blok (BBF). Transformasi jarak dan skema seleksi bobot adaptif diperkenalkan ke BBF untuk mencapai peningkatan akurasi dan kecepatan dalam proses pewarnaan.

Berdasarkan penelitian sebelumnya penulis berencana untuk melakukan penelitian yang berjudul Desain Sistem Pewarnaan Otomatis Berbasis Fitur Tekstur GLCM (Gray Level Co-Occurrence Matrix) Menggunakan Backpropagation. Diharapkan hasil penelitian ini dapat dijadikan acuan dalam penemuan bukti kriminalitas atau untuk mendeteksi sebuah penyakit yang di temukan oleh dokter.

II. METODE PENELITIAN



Gambar 1 Metode penelitian

Gambar 1 Menunjukkan metode penelitian yang digunakan, dimulai dengan mencari data berupa citra gray scale dan citra warna. Proses selanjutnya pada citra warna di lakukan preprocessing untuk menyamakan persepsi dengan citra grayscale. Setelah kedua citra dalam satu persepsi baru kemudian di cari nilai nilai fitur tekstur GLCM, Fitur tekstur yang di gunakan dalam penelitian ini ada lima fitur yaitu: *angular second moment (ASM)*, *kontras*, *inverse different moment (IDM)*, entropi dan korelasi.

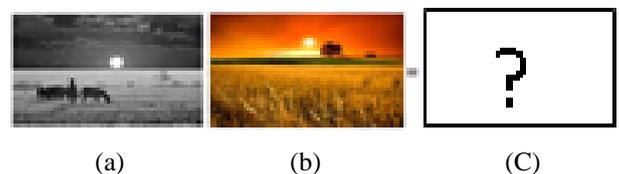
Kemudian di proses dengan model Backpropagation, pada model ini nilai-nilai fitur tekstur tersebut sebagai inputan yang nantinya akan di testing dan di training. Proses training dilakukan diluar aplikasi, yaitu dengan mengambil beberapa sampel data dan diproses sehingga nanti ditemukan bobot yang sesuai. Kemudian proses testing dilaksanakan dengan melakukan penanaman tahap

Feedforward dari jaringan syaraf tiruan Backpropagation pada aplikasi (Risaldi, Himawan, Teknik, Universitas, & Nuswantoro, 2014). sehingga menghasilkan nilai yang terbaik, nilai tersebut nanti nya digunakan untuk proses pewarnaan citra grayscale sampai dihasilkan sebuah citra warna

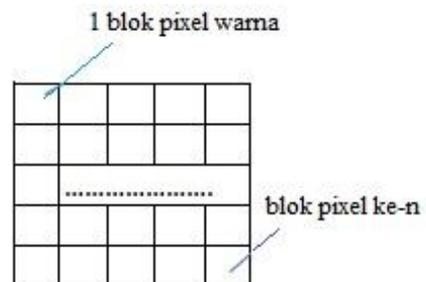
III. HASIL DAN PEMBAHASAN

3.1 Proses Desain

Proses desain diawali dengan mengambil data yaitu berupa citra grayscale yaitu citra yang akan di warnai dan citra warna sebagai citra acuan seperti yang terlihat pada gambar 3.1. Selanjutnya menentukan berapa ukuran piksel citra yang akan digunakan untuk mencari fitur GLCM, misal dalam penelitian ini blok-blok citra berukuran 10 x 10 piksel. blok citra tersebut ditunjukkan pada gambar 3.2 dibawah ini.



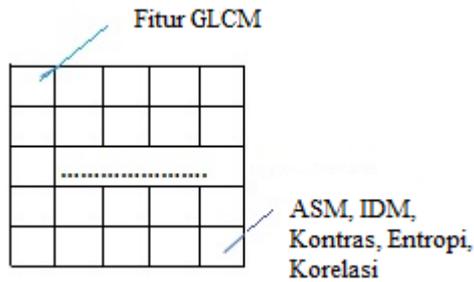
Gambar 2 Data Citra (a : Citra grayscale, b : citra warna , c : citra hasil Pewarnaan)



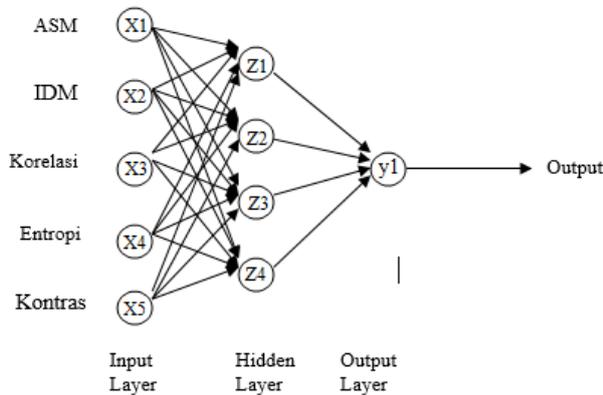
Gambar 3 Blok Piksel 10x10 Piksel (Sipan, N, & Yuniarno, 2017)

Proses selanjutnya menentukan nilai kelima fitur tekstur GLCM dari nilai ASM, IDM, Korelasi, Kontras dan Entropi (Malik & Baharudin, 2013). Nilai fitur yang diperoleh nantinya menjadi data input yang akan di proses

dengan jaringan syaraf tiruan (JST) backpropagation. Fitur blok citra ditunjukkan pada gambar 4



Gambar 4 Nilai Fitur GLCM Blok Citra



Gambar 5 Neural Network Backpropagation 1 Layer Hidden

Setelah data input didapatkan, selanjutnya menentukan layer dan neuron pada proses backpropagation seperti terlihat pada gambar 5. Lebih jelasnya rincian dari algoritma JST Backpropagation yang digunakan pada penelitian ini adalah sebagai berikut (Risaldi et al., 2014):

1. Algoritma pelatihan Backpropagation
 - Langkah 0
Inisialisasi penimbang (diberi nilai kecil secara acak)
 - Langkah 1
Ulangi langkah 2 sampai 9 sampai kondisi akhir iterasi dipenuhi
 - Langkah 2
Masing-masing pasangan data pelatihan (training data) lakukan langkah 3 sampai 8
2. Umpan maju (*FeedForward*)
 - Langkah 3
Pada masing-masing unit masukan (X_i $i=1 \dots n$) menerima sinyal masukkan X_i dan sinyal tersebut disebarkan ke unit-unit bagian berikutnya (unit-unit lapis tersembunyi)
 - Langkah 4
Kemudian masing-masing unit di layer tersembunyi dikalikan dengan penimbang, selanjutnya dijumlahkan dan ditambah dengan biasnya : n

$$Z_inj = Vj0 + \sum_{i=1}^n XiVji$$
 Kemudian dihitung sesuai dengan fungsi pengaktif yang digunakan:

$$Z_j = f(Z_inj)$$

Jika yang digunakan adalah fungsi sigmoid maka bentuk fungsi tersebut adalah:

$$Z_j = 1 + \exp(-z_inj)$$

Menghitung perbaikan penimbang (kemudian untuk memperbaiki wjk).

$$\Delta W_{kj} = \alpha \cdot k \dots Z_j$$

Menghitung perbaikan koreksi yang dilakukan:

$$\Delta W_{0j} = \alpha \cdot k$$

Kemudian menggunakan nilai delta (dk) pada semua unit lapis sebelumnya.

• Langkah 7

Masing-masing penimbang yang menghubungkan unit-unit lapis keluaran dengan unit-unit pada lapis tersembunyi (Z_j , $j=1, \dots, p$) dikalikan delta (δ) dan dijumlahkan sebagai masukan ke unit-unit lapis berikutnya.

$$\delta_inj = \sum_{k=1}^p \delta_i W_{ij}$$

Selanjutnya dikalikan dengan turunan dari fungsi pengaktifnya untuk mengitung galat.

$$\delta_j = \delta_inj f'(y_inj)$$

Langkah selanjutnya mengitung perbaikan penimbang (digunakan untuk memperbaiki V_{ij}).

$$\Delta V_{ij} = \alpha \delta_j \cdot X_i$$

Kemudian mengitung perbaikan bias (untuk memperbaiki V_{0j})

$$\Delta V_{0j} = \alpha \delta_j$$

3. Memperbaiki penimbang dan bias

• Langkah 8

Masing-masing keluaran unit (Y_k , $k=1, \dots, m$) diperbaiki bias dan penimbangnya ($j=0, \dots, p$).

$$W_{kj} \text{ (baru)} = W_{kj} \text{ (lama)} + \Delta W_{kj}$$

Masing-masing unit tersembunyi (Z_j , $j=1, \dots, p$) diperbaiki bias dan penimbangnya ($j=0, \dots, n$).

$$V_{ji} \text{ (baru)} = V_{ji} \text{ (lama)} + \Delta V_{ji}$$

• Langkah 9

Uji kondisi pemberhentian dengan menggunakan dua cara yaitu membatasi iterasi dan menentukan besar Mean Square Error antara output yang dikehendaki dan output yang dihasilkan.

$$MSE = 0,5 \times \{(t_1 - y_{k1})^2 + (t_2 - y_{k2})^2 + \dots + (t_m - y_{km})^2\}$$

Daftar Notasi :

X_p = Pola masukan pelatihan ke-p, $p=1, 2, \dots, p \leq 1$
 $X_p = (X_1, X_2, X_3, \dots, X_n)$

T_p = Pola target keluaran dari pelatihan

$t_p = (t_1, t_2, t_3, \dots, t_n)$

x_i = Unit ke-i pada lapisan masukan

X_i = Nilai pengaktif dari unit X_i

ΔW_{kj} = Selisih antara W_{kj} (t) dengan W_{kj} (t+1)
 V_{j0} = Nilai penimbang pada bias untuk unit

Z_j V_{ij} = Nilai penimbang dari unit X_i ke unit Z_j

ΔV_{ij} = Selisih antara V_{ij} (t) dengan V_{ij} (t+1)

Δk = Faktor pengendalian nilai penimbang pada lapis keluaran

δ_j = Faktor pengendalian nilai penimbang pada lapis tersembunyi

α = Konstanta laju pelatihan (learning rate)
 $0 < \alpha < 1$

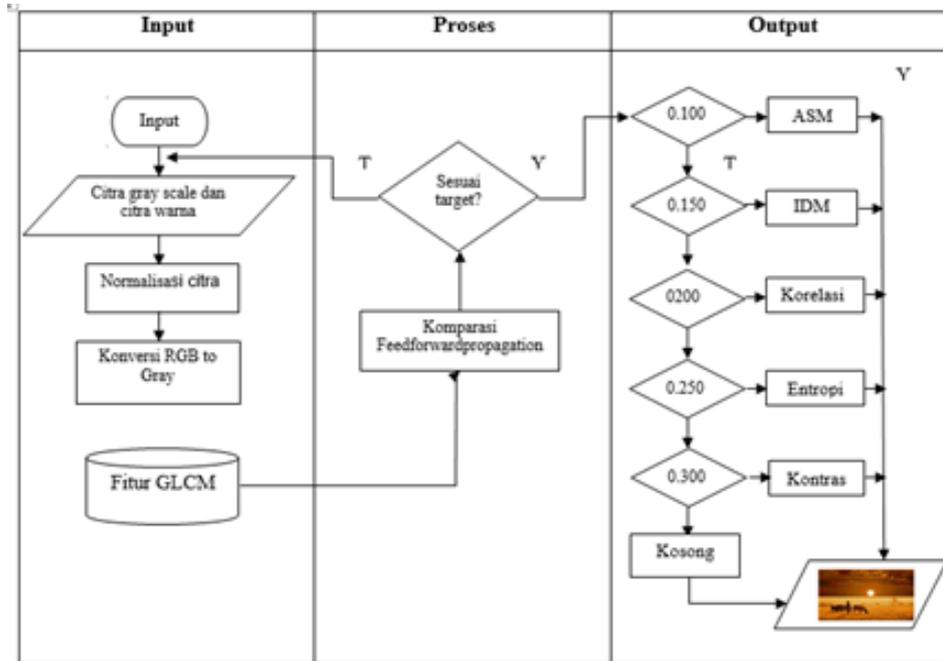
3.5 Desain Sistem

Hasil desain system penelitian ini ditunjukkan pada gambar 6 dibawah ini, desain terdiri dari input, proses dan output. Input desain meliputi dua citra yaitu citra grayscale yang akan diwarnai dan citra warna sebagai acuan pewarnaan. Sebelum citra warna di ubah ke grayscale untuk di cari fitur teksturnya maka kedua citra terlebih dahulu dinormalisasi, normalisasi citra merupakan mengganti jangkauan nilai intensitas piksel citra. Kegunaannya antara lain perbaikan citra berkontras rendah akibat terkena sinar berlebih.

backpropagation dapat memberikan gambaran yang lebih jelas dan terinci untuk melakukan sebuah pewarnaan citra secara otomatis.

UCAPAN TERIMA KASIH

Penelitian ini terlaksana atas dukungan LPPM Universitas Semarang sebagai penyandang dana dan Teknik Elektro pada khususnya, beserta tim peneliti, semoga dukungan dan partisipasi tim dan teman seperjuangan dapat memberikan semangat kita untuk lebih maju kedepanya



Gambar 6 Desain Sistem Pewarnaan Otomatis Berbasis Fitur Tekstur GLCM

Proses selanjutnya mengkonversi citra warna (citra acuan) menjadi citra grayscale, selanjutnya kedua citra grayscale (satu citra acuan dan satu citra yang akan diwarnai) dibuat blok-blok citra seperti gambar 5 di cari nilai fitur teksturnya yaitu berupa fitur tekstur GLCM. Fitur tekstur GLCM yang digunakan pada penelitian ini berjumlah lima fitur yang terdiri dari ASM, IDM, Kontras, Entropi dan Korelasi. Kelima fitur tekstur pada ratusan blok citra tersebut di testing dan di training sampai menghasilkan nilai error yang paling kecil atau dengan kata lain nilai keberhasilan yang paling tinggi. Nilai error yang paling kecil pada masing-masing fitur dan masing-masing blok dapat diasumsikan bahwa blok kedua citra tersebut memiliki warna yang sama. Setelah melalui proses *feedbackpropagation* maka warna blok citra warna ditransfer ke blok citra grayscale sampai blok citra grayscale terwarnai semua sehingga citra grayscale tersebut menjadi citra warna ditunjukkan pada output desain sistem gambar 6.

IV. SIMPULAN

Berdasarkan penelitian yang kita lakukan, dapat diambil kesimpulan bahwa. Sebuah model system pewarnaan citra otomatis berbasis fitur tekstur glcm (gray level co-occurrence matrix) menggunakan

DAFTAR PUSTAKA

- Alekseev, A., Rozaliev, V., & Orlova, Y. (2015). Automization Colorize Grayscale Images Based Intelligent Scene Analysis. *Springer, Recognition and Image Analysis*, 2015, 3–6.
- Alexandru, C., Ivanovici, M., & Richard, N. (2014). *Probabilistic pseudo-morphology for grayscale and color images*. \$ 47, 721–735. <https://doi.org/10.1016/j.patcog.2013.08.021>
- Hogervorst, M. A., & Toet, A. (2010). Fast natural color mapping for night-time imagery. *Information Fusion*, 11(2), 69–77. <https://doi.org/10.1016/j.inffus.2009.06.005>
- Kolmogorov, V., & Rother, C. (2007). *Minimizing Nonsubmodular Functions with Graph Cuts — A Review*. 29(7), 1274–1279.
- Liu, S., & Zhang, X. (2012). Automatic grayscale image colorization using histogram regression. *Pattern Recognition Letters*, 33(13), 1673–1681. <https://doi.org/10.1016/j.patrec.2012.06.001>
- Malik, F., & Baharudin, B. (2013). *The Statistical Quantized Histogram Texture Features Analysis for Image Retrieval Based on Median and Laplacian Filters in the DCT Domain*. 10(6).

- Risaldi, M., Himawan, H., Teknik, P., Universitas, I., & Nuswantoro, D. (2014). *KLASIFIKASI KUALITAS KAYU KELAPA MENGGUNAKAN ALGORITMA NEURAL NETWORK BACKPROPAGATION*. 10(April), 69–78.
- Sipan, M., N, S. M. S., & Yuniarno, E. M. (2017). *Image Block Matching Based on GLCM (Gray level Co-occurrence Matrix) Texture Feature on Grayscale Image Auto Coloring*.
- Sousa, A., & Blaes, P. (n.d.). *Automatic Colorization of Grayscale Images Colorization*. 2–6.