

IMPLEMENTASI SELF ORGANIZING MAP DALAM KOMPRESI CITRA DIGITAL

Hisar M. Simbolon⁽¹⁾
hisarliska@gmail.com

Sri Suwarno⁽²⁾
sswn@ukdw.ac.id

Restyandito⁽³⁾
dito@ukdw.ac.id

Abstraksi

Kompresi citra digital merupakan suatu bagian yang tidak dapat dipisahkan dari perkembangan sistem informasi. Seiring dengan perkembangan transmisi data dibutuhkan ukuran file gambar yang kecil sehingga dapat ditransmisikan dengan kecepatan yang semaksimalnya.

Solusi dari permasalahan adalah dengan membangun aplikasi kompresi citra digital dengan menerapkan algoritma Self – Organizing Map (SOM) untuk mengolah tiap – tiap piksel pada citra sehingga kompresi dapat dimaksimalkan dengan Run Length Encode. Hasil dekompresi yang tidak jauh berbeda dengan citra asli merupakan hasil yang sangat diharapkan agar tidak memberikan arti yang berbeda terhadap citra asli.

Kesimpulan yang diperoleh dari penelitian ini adalah penggunaan algoritma SOM dapat mengolah setiap piksel pada citra digital dan mengurangi redundansi citra sehingga algoritma Run Length dapat dengan maksimal mengodekan setiap warna pada piksel hingga mencapai tingkat rasio kompresi 60% - 85%. Hasil dekompresi citra mengalami penurunan kualitas gambar namun masih mengandung data - data yang sama atau dengan kata lain gambar dekompresi tidak jauh berbeda dengan gambar asli.

Kata kunci : Algoritma SOM, Run length, kompresi.

1. Pendahuluan

Sebuah citra dapat digunakan untuk berbagai kepentingan baik sebagai dokumentasi maupun sebagai alat penyelarasan dalam sebuah aplikasi. Pemanfaatan sebuah citra semakin penting di dalam sistem informasi terlebih saat citra tersebut ingin didokumentasikan ke dalam bentuk digital.

¹ Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana

² Teknik Informatika, Fakultas Teknologi Informasi Univeristas Kristen Duta Wacana

³ Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana

Permasalahan yang dihadapi adalah ukuran data yang semakin besar karena semakin banyak dokumentasi citra. Hal ini akan sangat berpengaruh pada media penyimpanan data yaitu *harddisk*, walaupun *harddisk* memiliki kapasitas yang besar tetapi akan tetap memiliki keterbatasan dengan sebuah alasan yaitu perbandingan antara harga dari *harddisk* dengan nilai citra yang akan disimpan. Saat citra akan ditransmisikan dalam suatu komunikasi jaringan terdapat juga permasalahan dalam prosesnya, yaitu semakin besar ukuran citra maka akan semakin lama proses transmisinya.

Melihat permasalahan tersebut dibutuhkan suatu solusi yang dapat mengatasi masalah ukuran citra. Kompresi citra merupakan salah satu solusi yang dapat membantu agar ukuran citra tersebut lebih kecil yang kemudian citra dapat ditransmisikan lebih cepat. Kompresi citra dapat dilakukan dengan berbagai cara, salah satunya adalah dengan metode *Self Organizing Map* (SOM). Metode SOM berfungsi untuk mengelompokkan beberapa data warna yang memiliki kemiripan dan akan dijadikan menjadi satu kelompok sehingga dapat dikompres lebih maksimal.

Dengan adanya sistem kompresi citra ini diharapkan bisa memperkecil ukuran citra sehingga citra tetap dapat disimpan lebih banyak lagi dan proses transmisinya juga dapat lebih cepat.

2. Landasan Teori

2.1. Metode Self-Organizing Map (SOM)

Pada prinsipnya algoritma SOM mempunyai 2 proses perhitungan matematika, yaitu pada proses pencarian nilai bobot yang sesuai dengan nilai masukan dan perubahan nilai bobot yang telah ditemukan dengan jarak terdekat.

Perhitungan perubahan nilai bobot sekitar atau tetangga bobot pemenang tidak dihitung atau diberi nilai 0. Pemberian nilai ini dimaksudkan agar tiap bobot diarahkan ke nilai masukan sehingga nilai bobot akan mendekati nilai masukan.

Di tahun 2005, Siang menuliskan algoritma SOM :

- Step 1 : Neuron pada lapisan input (neuron input) sebanyak n dinotasikan sebagai x_1, x_2, \dots, x_n dan neuron pada lapisan output (neuron output) sebanyak m dinotasikan sebagai y_1, y_2, \dots, y_m . Bobot koneksi antara neuron input dan output dinotasikan sebagai w_{ij} ditentukan secara acak antara 0 dan 1.
- Step 2 : Selama kondisi penghentian bernilai salah, lakukan langkah 3 – 8
- Step 3 : Untuk setiap masukan (x_1, x_2, \dots, x_n) lakukan langkah 4 – 6

Step 4 : Hitung jarak vektor input terhadap bobot koneksi d_j untuk masing-masing neuron output dengan menggunakan rumus:

$$d_j = \sum_{i=1}^n (w_{ij} - x_i)^2 \quad [1]$$

Step 5 : Cari indeks j di mana d_j minimum

Step 6 : Untuk setiap w_{ij} , perbaharui bobot koneksi dengan menggunakan rumus:

$$w_{ij}(t+1) = w_{ij}(t) + \gamma(t)h_{ib}(t) (x_i(t) - w_{ij}(t))$$

Step 7 : Modifikasi Laju Pemahaman

Step 8 : Uji kondisi penghentian

2.2. Metode Run – Length Encoding

Penyusunan data kompresi citra dengan *run length encoding* membutuhkan sebuah penanda antara data citra dan banyaknya data citra. Penanda ini bertujuan untuk mengetahui dimana yang merupakan data dan yang mana yang merupakan banyaknya data.

Langkah-langkah yang dilakukan dalam proses *run length encoding* sebagai berikut :

1. Lihat apakah terdapat deretan data yang sama secara berurutan lebih dari tiga karakter, jika memenuhi lakukan kompresi.
2. Berikan bit penanda pada data yang dikompres, bit penanda disini berupa 8 deretan bit yang boleh dipilih sembarang asalkan digunakan secara konsisten pada seluruh bit penanda dalam kompresi. Bit penanda ini berfungsi untuk menandai bahwa karakter selanjutnya adalah karakter pemampatan sehingga tidak membingungkan pada saat mengembalikan data yang sudah dimampatkan ke file aslinya.
3. Tambahkan deretan bit untuk menyatakan jumlah karakter yang sama berurutan.
4. Tambahkan deretan bit yang menyatakan data yang berulang.

2.3. Metode Run – Length Decoding

Proses pengembalian data yang dikodekan menjadi data asli disebut *decoding*. Algoritma *run length* merubah suatu data ke dalam bentuk kode dengan aturan – aturan yang sudah dijelaskan di atas. Oleh sebab itu dibutuhkan suatu algoritma untuk mengembalikan data kode menjadi data asli sehingga data tersebut bisa digunakan sebagaimana mestinya.

Run length decoding berfungsi sebagai kebalikan dari *run length encoding* yaitu merubah data kode menjadi data asli. Langkah – langkah untuk mengembalikan kode dimaksud adalah sebagai berikut :

1. Lihat karakter pada hasil kompresi satu – persatu dari awal sampai akhir, jika ditemukan bit penanda, lakukan proses pengembalian.
2. Lihat karakter setelah bit penanda, karakter setelah bit penanda merepresentasikan nilai jumlah karakter yang sama.
3. Lihat karakter berikutnya (karakter ke-2 setelah bit penanda), kemudian lakukan penulisan karakter tersebut sebanyak bilangan yang telah diperoleh pada karakter sebelumnya (point 2).

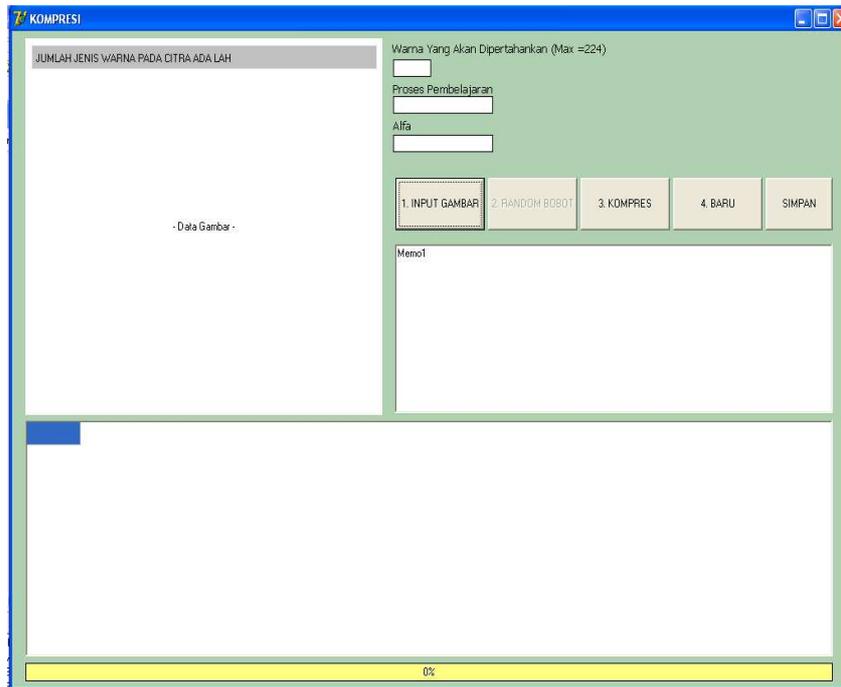
3. Pembahasan dan Hasil

Pada program utama terdapat pilihan menu KOMPRESI. Aplikasi ini merupakan implementasi algoritma Jaringan Syaraf Tiruan (JST) dengan metode *Self – Organizing Map* (SOM) dan *Run Length*. Hasil dari menu ini merupakan kode kompresi dengan ber-ekstensi .esi (dot esi).

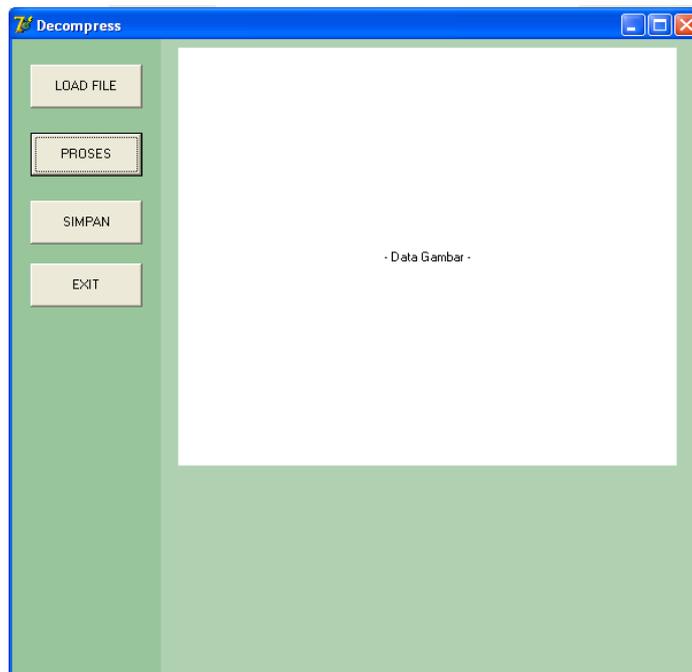
Program kompresi ini membutuhkan beberapa masukan yang akan dibutuhkan untuk menjalankan program kompresi. Masukan – masukan tersebut antara lain :

- Warna yang akan dipertahankan : masukan ini berfungsi sebagai pembentuk matrik yang akan dibentuk program.
- Proses Pembelajaran : masukan ini dibutuhkan untuk algoritma SOM sehingga proses berhentinya pelatihan dapat dibatasi.
- Alfa : masukan ini dibutuhkan untuk algoritma SOM sebagai laju pembelajaran, nilainya berupa nilai decimal dengan titik sebagai pemisah bilangan.

Pada program kompresi juga terdapat tombol – tombol yang di *click* secara berurutan mulai dari INPUT GAMBAR, RANDOM BOBOT, KOMPRESI. Ketiga tombol tersebut dioperasikan secara berurutan untuk menyesuaikan dengan algoritma SOM. tombol BARU dan tombol SIMPAN dapat digunakan setelah proses kompresi selesai. Tampilan kompresi dapat dilihat pada Gambar 1.



Gambar 1. Menu Kompresi



Gambar 2. Menu Dekompresi

Menu berikutnya yang disediakan pada menu utama adalah menu DEKOMPRESI. Menu dekomposisi berfungsi untuk mengembalikan kode kompresi menjadi gambar. Pada proses dekomposisi hanya membutuhkan masukan kode kompresi saja, selanjutnya hanya *click* tombol – tombol untuk melakukan proses berikutnya, tombol PROSES untuk dekomposisi dan tombol SIMPAN untuk menyimpan hasil dekomposisi gambar. Tampilan menu dekomposisi dapat dilihat pada Gambar 2.

3.1. Analisis Hasil Kompresi Dan Dekompresi

Secara teori semakin sering seseorang belajar maka akan semakin ingat atau hafal akan apa yang dia pelajari. Hal ini sangat umum sekali, namun pada pengujian ini akan diteliti terhadap proses pembelajaran yang kecil hingga yang lebih besar dan melihat dampak yang akan memberikan terhadap besar ukuran kompresi. Hasil pengujian yang telah dilakukan dapat dilihat pada Table 1.

Table 1. Hasil Pengujian Dengan Masukan Proses Pembelajaran Yang Berbeda

Proses Pembelajaran	Alfa (<i>Learning Rate</i>)	Besar Citra Asli (<i>Byte</i>)	Besar Hasil Kompresi (<i>Byte</i>)	Rasio Kompresi	Lama Proses Kompresi
100	0.02	30054	8061	0.2572702	00:10:12:953
500	0.02	30054	7954	0.2788979	00:52:37:125
1000	0.02	30054	8136	0.2635256	01:16:45:485
5000	0.02	30054	7786	0.2802289	03:27:15:512
10000	0.02	30054	7955	0.2553736	07:00:52:795

Perolehan hasil kompresi sebuah citra bola.bmp yang diuji dengan masukan seperti pada Table 1 tidak menunjukkan banyak perbedaan dari besar kompresi dan rasio kompresi, namun waktu proses kompresi menunjukkan semakin besar proses pembelajaran maka semakin lama juga proses kompresinya.

Perbedaan besar ukuran hasil kompresi tidak mengalami pengecilan namun semakin besar pada saat – saat tertentu. Misalnya dari proses pembelajaran 150 terhadap proses pembelajaran 200 rasio kompresi mengalami peningkatan yaitu semakin kecil rasio kompresi sedangkan sebelumnya dari proses pembelajaran 100 ke proses pembelajaran 150 justru mengalami penurunan yaitu semakin besar nilai rasio kompresi yang dihasilkan. Oleh karena itu dapat diperoleh bahwa proses pembelajaran tidak mempengaruhi besar kecilnya ukuran hasil kompresi.

3.2. Pengujian Gambar Hasil Dekompresi Dengan Gambar Asli

Hasil dekompresi merupakan pengembalian hasil kompresi menjadi sebuah citra kembali. Pada bagian ini akan ditampilkan hasil gambar yang telah diujikan terhadap system. Akan ditampilkan juga gambar asli sebagai perbandingan yang dapat dilihat secara *visual*.

Hasil dekompresi kode kompresi menjadi sebuah citra juga menjadi pertimbangan yang penting, karena dengan meng-kompresi sebuah gambar bola maka diharapkan juga hasil dekompresi menghasilkan sebuah gambar bola dan sama seperti asli. Kemiripan gambar hasil dekompresi dengan gambar asli merupakan suatu parameter untuk pengukuran gagal atau suksesnya metode tersebut dalam kompresi citra



Gambar 3. Hasil Dekompresi

4. Kesimpulan

Program aplikasi kompresi citra dengan menggunakan jaringan Kohonen *Self – Organizing Map* dapat melakukan kompresi dengan baik. Pengembalian hasil kompresi menjadi sebuah citra juga menghasilkan gambar yang cukup mirip dengan gambar asli walaupun dengan menggantikan warna piksel dengan warna yang memiliki nilai kemiripan paling dekat dengan warna sekitarnya.

Kesimpulan yang dapat diperoleh dari program kompresi ini adalah sebagai berikut :

- a) Nilai proses pembelajaran mempunyai pengaruh yang sangat tinggi terhadap hasil dekompresi gambar, semakin tinggi nilai proses pembelajaran maka hasil dekompresi akan semakin mirip dengan gambar asli.
- b) Semakin tinggi nilai proses pembelajaran maka waktu yang dibutuhkan untuk melakukan kompresi akan semakin lama.
- c) Nilai laju pemahaman yang terlalu rendah dengan proses pembelajaran yang sedikit akan memberikan hasil dekompresi yang tidak bagus.

Daftar Pustaka

Hermawan, A. (2006). *Jaringan Saraf Tiruan Teori dan Aplikasi*. Yogyakarta : ANDI.

Kusumadewi, S. (2004). *Membangun Jaringan Syaraf Tiruan Menggunakan MATLAB & EXCEL LINK*. Yogyakarta : GRAHA ILMU.

Salomon, D. (2004). *Data Compression 3rd Edition*. New York: Springer-Verlag

Siang, J.J., (2004). *Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan Matlab*. Yogyakarta : ANDI.