

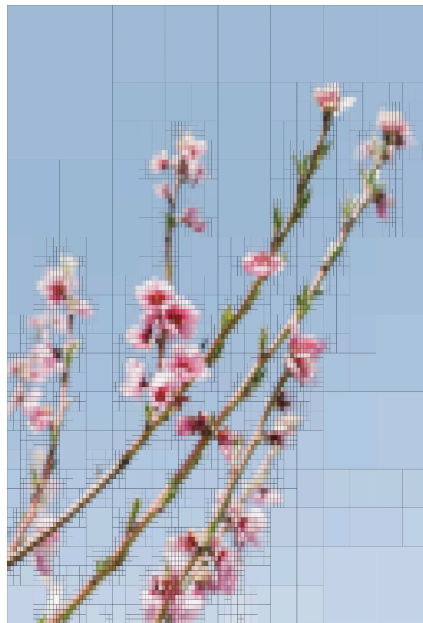
Tugas Kecil 2 IF2211 Strategi Algoritma
Semester II tahun 2024/2025
Kompresi Gambar Dengan Metode Quadtree

Batas pengumpulan : Hari Jumat, 11 April 2025 pukul 23.59 WIB

Arsip pengumpulan :

- *Source* program yang dapat dijalankan disertai README
- Laporan (*soft copy*)

Deskripsi Tugas:



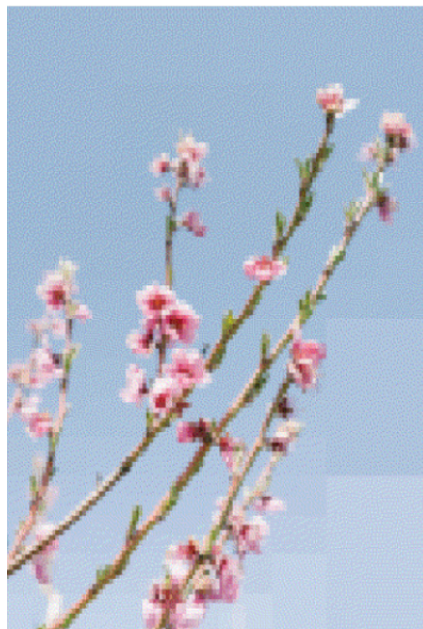
Gambar 1. Quadtree dalam Kompresi Gambar

(Sumber: <https://medium.com/@tannerwyork/quadtrees-for-image-processing-302536c95c00>)

Quadtree adalah struktur data hierarkis yang digunakan untuk membagi ruang atau data menjadi bagian yang lebih kecil, yang sering digunakan dalam pengolahan gambar. Dalam konteks kompresi gambar, Quadtree membagi gambar menjadi blok-blok kecil berdasarkan keseragaman warna atau intensitas piksel. Prosesnya dimulai dengan membagi gambar menjadi empat bagian, lalu memeriksa apakah setiap bagian memiliki nilai yang seragam berdasarkan analisis **sistem warna RGB**, yaitu dengan membandingkan komposisi nilai merah (R), hijau (G), dan biru (B) pada piksel-piksel di dalamnya. Jika bagian tersebut tidak seragam, maka bagian

tersebut akan terus dibagi hingga mencapai tingkat keseragaman tertentu atau ukuran minimum yang ditentukan.

Dalam implementasi teknis, sebuah Quadtree direpresentasikan sebagai simpul (node) dengan maksimal empat anak (children). Simpul daun (leaf) merepresentasikan area gambar yang seragam, sementara simpul internal menunjukkan area yang masih membutuhkan pembagian lebih lanjut. Setiap simpul menyimpan informasi seperti posisi (x, y), ukuran (width, height), dan nilai rata-rata warna atau intensitas piksel dalam area tersebut. Struktur ini memungkinkan pengkodean data gambar yang lebih efisien dengan menghilangkan redundansi pada area yang seragam. QuadTree sering digunakan dalam algoritma kompresi lossy karena mampu mengurangi ukuran file secara signifikan tanpa mengorbankan detail penting pada gambar.



Gambar 2. Proses Pembentukan Quadtree dalam Kompresi Gambar

(Sumber: https://miro.medium.com/v2/resize:fit:640/format:webp/1*LHD7PsbmbgNBFrYkxyG5dA.gif)

****Cek sumber untuk melihat animasi GIF****

Ilustrasi kasus :

Ide pada tugas kecil 2 ini cukup sederhana, seperti pada pembahasan sebelumnya mengenai Quadtree. Berikut adalah prosedur pada program kompresi gambar yang akan dibuat dalam Tugas Kecil 2 (*Divide and Conquer*):

1. Inisialisasi dan Persiapan Data

Masukkan gambar yang akan dikompresi akan diolah dalam format matriks piksel dengan nilai intensitas berdasarkan sistem warna RGB. Berikut adalah parameter-parameter yang dapat ditentukan oleh pengguna saat ingin melakukan kompresi gambar:

- a) **Metode perhitungan variansi:** pilih metode perhitungan variansi berdasarkan opsi yang tersedia pada *Tabel 1*.
- b) **Threshold variansi:** nilai ambang batas untuk menentukan apakah blok akan dibagi lagi.
- c) **Minimum block size:** ukuran minimum blok piksel yang diperbolehkan untuk diproses lebih lanjut.

2. Perhitungan Error

Untuk setiap blok gambar yang sedang diproses, hitung nilai variansi menggunakan metode yang dipilih sesuai *Tabel 1*.

3. Pembagian Blok

Bandingkan nilai variansi blok dengan threshold:

- ❖ Jika variansi **di atas threshold** (*cek kasus khusus untuk metode bonus*), ukuran blok lebih besar dari **minimum block size**, dan ukuran blok setelah dibagi menjadi empat **tidak kurang dari minimum block size**, blok tersebut dibagi menjadi empat sub-blok, dan proses dilanjutkan untuk setiap sub-blok.
- ❖ Jika salah satu kondisi di atas tidak terpenuhi, proses pembagian dihentikan untuk blok tersebut.

4. Normalisasi Warna

Untuk blok yang tidak lagi dibagi, lakukanlah normalisasi warna blok sesuai dengan rata-rata nilai RGB blok.

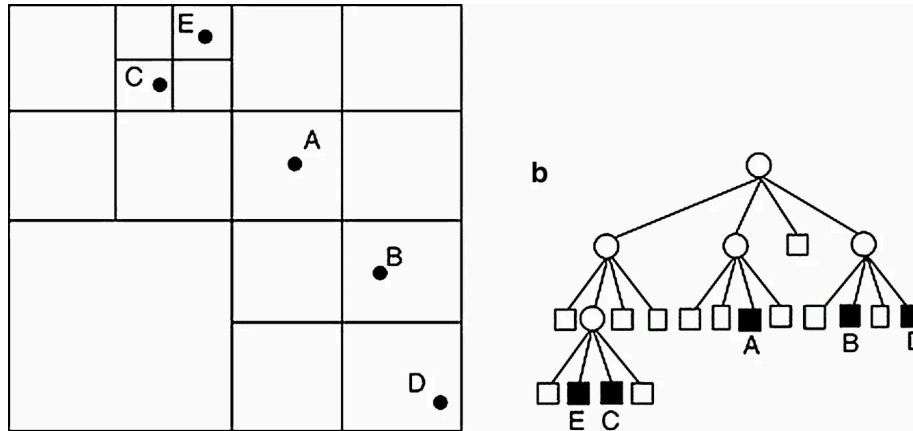
5. Rekursi dan Penghentian

Proses pembagian blok dilakukan secara rekursif untuk setiap sub-blok hingga semua blok memenuhi salah satu dari dua kondisi berikut:

- ❖ Error blok berada di bawah threshold.
- ❖ Ukuran blok setelah dibagi menjadi empat kurang dari *minimum block size*.

6. Penyimpanan dan Output

Rekonstruksi gambar dilakukan berdasarkan struktur QuadTree yang telah dihasilkan selama proses kompresi. Gambar hasil rekonstruksi akan disimpan sebagai file terkompresi. Selain itu, persentase kompresi akan dihitung dan disertakan dengan rumus sesuai dengan yang terlampir pada dokumen ini. Persentase kompresi ini memberikan gambaran mengenai efisiensi metode kompresi yang digunakan.



Gambar 3. Struktur Data Quadtree dalam Kompresi Gambar

(Sumber: <https://medium.com/@tannerwyork/quadtrees-for-image-processing-302536c95c00>)

Parameter:

1. Error Measurement Methods (Metode Pengukuran Error)

Metode yang digunakan untuk menentukan seberapa besar perbedaan dalam satu blok gambar. Jika error dalam blok melebihi ambang batas (*threshold*), maka blok akan dibagi menjadi empat bagian yang lebih kecil.

Tabel 1. Metode Pengukuran Error

Metode	Formula
<u>Variance</u>	$\sigma_c^2 = \frac{1}{N} \sum_{i=1}^N (P_{i,c} - \mu_c)^2$
	$\sigma_{RGB}^2 = \frac{\sigma_R^2 + \sigma_G^2 + \sigma_B^2}{3}$
	σ_c^2 = Variansi tiap kanal warna c (R, G, B) dalam satu blok
	$P_{i,c}$ = Nilai piksel pada posisi i untuk kanal warna c
	μ_c = Nilai rata-rata tiap piksel dalam satu blok
	N = Banyaknya piksel dalam satu blok

Mean Absolute Deviation (MAD)	$MAD_c = \frac{1}{N} \sum_{i=1}^N P_{i,c} - \mu_c $
	$MAD_{RGB} = \frac{MAD_R + MAD_G + MAD_B}{3}$
	MAD_c = Mean Absolute Deviation tiap kanal warna c (R, G, B) dalam satu blok $P_{i,c}$ = Nilai piksel pada posisi i untuk kanal warna c μ_c = Nilai rata-rata tiap piksel dalam satu blok N = Banyaknya piksel dalam satu blok
Max Pixel Difference	$D_c = \max(P_{i,c}) - \min(P_{i,c})$
	$D_{RGB} = \frac{D_R + D_G + D_B}{3}$
	D_c = Selisih antara piksel dengan nilai max dan min tiap kanal warna c (R, G, B) dalam satu blok $P_{i,c}$ = Nilai piksel pada posisi i untuk channel warna c
Entropy	$H_c = - \sum_{i=1}^N P_c(i) \log_2(P_c(i))$
	$H_{RGB} = \frac{H_R + H_G + H_B}{3}$
	H_c = Nilai entropi tiap kanal warna c (R, G, B) dalam satu blok $P_c(i)$ = Probabilitas piksel dengan nilai i dalam satu blok untuk tiap kanal warna c (R, G, B)
[Bonus] Structural Similarity Index (SSIM) (Referensi tambahan)	$SSIM_c(x, y) = \frac{(2\mu_{x,c}\mu_{y,c} + C_1)(2\sigma_{xy,c} + C_2)}{(\mu_{x,c}^2 + \mu_{y,c}^2 + C_1)(\sigma_{x,c}^2 + \sigma_{y,c}^2 + C_2)}$
	$SSIM_{RGB} = w_R \cdot SSIM_R + w_G \cdot SSIM_G + w_B \cdot SSIM_B$
	<p>Nilai SSIM yang dibandingkan adalah antara blok gambar sebelum dan sesudah dikompresi. Silakan lakukan eksplorasi untuk memahami serta memperoleh nilai konstanta pada formula SSIM, asumsikan gambar yang akan diuji adalah 24-bit RGB dengan 8-bit per kanal.</p>

2. Threshold (Ambang Batas)

Threshold adalah nilai batas yang menentukan apakah sebuah blok dianggap cukup seragam untuk disimpan atau harus dibagi lebih lanjut.

3. Minimum Block Size (Ukuran Blok Minimum)

Minimum block size (luas piksel) adalah ukuran terkecil dari sebuah blok yang diizinkan dalam proses kompresi. Jika ukuran blok yang akan dibagi menjadi empat sub-blok berada di bawah ukuran minimum yang telah dikonfigurasi, maka blok tersebut tidak akan dibagi lebih lanjut, meskipun error masih di atas threshold.

4. Compression Percentage (Persentase Kompresi) [BONUS]

Persentase kompresi menunjukkan seberapa besar ukuran gambar berkurang dibandingkan dengan dengan ukuran aslinya setelah dikompresi menggunakan metode quadtree.

$$\text{Persentase Kompresi} = \left(1 - \frac{\text{Ukuran Gambar Terkompresi}}{\text{Ukuran Gambar Asli}}\right) \times 100\%$$

Spesifikasi Tugas Kecil 2:

- Buatlah program sederhana dalam bahasa **C/C#/C++/Java** (CLI) yang mengimplementasikan **algoritma *divide and conquer*** untuk melakukan kompresi gambar berbasis *quadtree* yang mengimplementasikan **seluruh parameter** yang telah disebutkan sebagai *user input*.
- Alur program:
 1. [INPUT] **alamat absolut** gambar yang akan dikompresi.
 2. [INPUT] metode perhitungan error (gunakan penomoran sebagai *input*).
 3. [INPUT] ambang batas (pastikan *range* nilai sesuai dengan metode yang dipilih).
 4. [INPUT] ukuran blok minimum.
 5. [INPUT] Target persentase kompresi (*floating number*, 1.0 = 100%), beri nilai 0 jika ingin menonaktifkan mode ini, jika mode ini aktif maka nilai threshold bisa menyesuaikan secara otomatis untuk memenuhi target persentase kompresi (bonus).
 6. [INPUT] **alamat absolut** gambar hasil kompresi.
 7. [INPUT] **alamat absolut** gif (bonus).

8. [OUTPUT] waktu eksekusi.
 9. [OUTPUT] ukuran gambar sebelum.
 10. [OUTPUT] ukuran gambar setelah.
 11. [OUTPUT] persentase kompresi.
 12. [OUTPUT] kedalaman pohon.
 13. [OUTPUT] banyak simpul pada pohon.
 14. [OUTPUT] gambar hasil kompresi pada alamat yang sudah ditentukan.
 15. [OUTPUT] GIF proses kompresi pada alamat yang sudah ditentukan (bonus).
- Berkas laporan yang dikumpulkan adalah laporan dalam bentuk PDF yang setidaknya berisi:
 1. Algoritma ***divide and conquer*** yang digunakan, jelaskan langkah-langkahnya, **bukan hanya notasi pseudocode dan BUKAN IMPLEMENTASINYA melainkan algoritmanya.**
 2. *Source* program dalam bahasa pemrograman yang dipilih (pastikan bahwa program telah dapat dijalankan).
 3. Tangkapan layar yang memperlihatkan *input* dan *output* (minimal sebanyak 7 buah contoh).
 4. Hasil analisis percobaan algoritma ***divide and conquer*** dalam kompresi gambar dengan metode Quadtree. Analisis dilakukan dalam bentuk paragraf/poin dan minimal memuat mengenai analisis kompleksitas algoritma program yang telah dikembangkan.
 5. Penjelasan mengenai implementasi bonus jika mengerjakan.
 6. Pranala ke *repository* yang berisi kode program.
 - **BONUS:**

Pastikan sudah mengerjakan spesifikasi wajib sebelum mengerjakan bonus:

 1. Memungkinkan pengguna menentukan **target persentase kompresi** berupa nilai floating point (contoh: 1.0 = 100%). Jika pengguna memberikan nilai 0, mode ini akan dinonaktifkan. Ketika mode ini diaktifkan, algoritma akan menyesuaikan nilai threshold secara otomatis untuk mencapai target persentase kompresi yang ditentukan. Penyesuaian threshold ini dilakukan secara dinamis, memberikan fleksibilitas dalam proses kompresi untuk memenuhi target efisiensi sambil mempertahankan kualitas gambar.
 2. Implementasikan **Structural Similarity Index (SSIM)** sebagai metode pengukuran error. Hitung SSIM untuk setiap kanal warna R, G, dan B, lalu gabungkan hasilnya menjadi SSIM total menggunakan bobot masing-masing kanal. Informasi mengenai formula perhitungan dan parameter SSIM dapat ditemukan di *Tabel 1*.

3. Membuat **GIF** proses kompresi seperti yang tertera pada **Gambar 2**. Visualisasi proses Pembentukan Quadtree dalam Kompresi Gambar dengan format **GIF**.
- Program disimpan dalam repository yang bernama Tucil2_NIM jika mengerjakan secara individu atau Tucil2_NIM1_NIM2 jika dikerjakan berkelompok. Berikut merupakan struktur dari isi repository tersebut:
 1. Folder **src** berisi *source code* program.
 2. Folder **bin** berisi *executable file* (Sesuaikan dengan bahasa pemrograman yang digunakan).
 3. Folder **test** berisi solusi jawaban dari data uji yang digunakan dalam laporan.
 4. Folder **doc** berisi laporan tugas kecil dalam bentuk PDF.
 5. **README** yang minimal berisi:
 - a. Penjelasan singkat program yang dibuat.
 - b. Requirement program dan instalasi tertentu bila ada.
 - c. Cara mengkompilasi program bila perlu dikompilasi (pastikan dengan langkah yang **jelas** dan **benar**).
 - d. Cara menjalankan dan menggunakan program (pastikan dengan langkah yang **jelas** dan **benar**).
 - e. Author / identitas pembuat.
 - Laporan dikumpulkan hari **Jumat, 11 April 2025** pada alamat Google Form berikut paling lambat pukul **23.59 WIB** (sebelum kelas kuliah):
<https://forms.gle/1x2DbnRJn3Zs9YzF7>
 - Pertanyaan terkait tugas kecil ini bisa disampaikan melalui QnA berikut:
<https://bit.ly/QnA-Stima-25>

Perhatikan:

- **Dilarang keras copy paste** program dari internet, AI, repository lain, ataupun teman. Program harus dibuat sendiri, kecurangan akan mengakibatkan nilai tugas menjadi **nol**.
- Pastikan program **dapat setidaknya dikompilasi** pada *windows* dan *linux*.
- Apabila program **tidak dapat dijalankan** maka tidak akan dinilai oleh asisten.
- Tugas dikerjakan oleh maksimal dua orang.
- Tambahkan tabel berikut yang diisi *checklist* (✓) pada bagian lampiran laporan Anda.

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalahan		
2. Program berhasil dijalankan		

3. Program berhasil melakukan kompresi gambar sesuai parameter yang ditentukan		
4. Mengimplementasi seluruh metode perhitungan error wajib		
5. [Bonus] Implementasi persentase kompresi sebagai parameter tambahan		
6. [Bonus] Implementasi Structural Similarity Index (SSIM) sebagai metode pengukuran error		
7. [Bonus] Output berupa GIF Visualisasi Proses pembentukan Quadtree dalam Kompresi Gambar		
8. Program dan laporan dibuat (kelompok) sendiri		

--- Selamat Mengerjakan! ---

“Selamat Lebaran guys 👍”

--- Asisten ---