



PENGEMBANGAN ALGORITMA PENGUBAHAN UKURAN CITRA BERBASISKAN ANALISIS GRADIEN DENGAN PENDEKATAN POLINOMIAL

Eric Christopher^{#1}, Dr. Ir. Rinaldi Munir, M. T.^{#2}

*#School of Electrical Engineering and Informatics, Institute Technology of Bandung,
10th Ganeca Street, Bandung, Indonesia.*

[1eric.c13@gmail.com](mailto:eric.c13@gmail.com)

[2rinaldi@informatika.org](mailto:rinaldi@informatika.org)

Abstrak— Dalam pengolahan citra, salah satu proses yang sering dilakukan adalah pengubahan ukuran citra. Salah satu kelemahan dari pengecilan suatu citra adalah apabila dilakukan perkecilan terhadap ukuran suatu citra bitmap maka informasi yang terkandung pada citra tersebut akan berkurang atau menghilang. Maka dari itu perlu digunakan suatu algoritma untuk menerka informasi-informasi yang hilang tersebut apabila citra tadi akan diperbesar lagi.

Dalam penelitian ini dikembangkan sebuah algoritma pengubahan ukuran citra dengan modifikasi dari algoritma bilinear yang merupakan salah satu algoritma umum yang dipakai dalam pengubahan ukuran citra. Dengan memanfaatkan fungsi polinomial diterka nilai pixel-pixel yang tadinya tidak diketahui. Fungsi yang digunakan menerapkan analisis gradien untuk memperjelas batas-batas antar objek dalam citra.

Perangkat lunak yang dibuat pada tugas akhir ini dibuat dengan bahasa java. Perangkat lunak ini mampu mengubah ukuran sebuah citra dengan algoritma yang dipilih. Perangkat lunak ini juga menyediakan fasilitas untuk membandingkan beberapa hasil perbesaran dari algoritma pengubahan citra yang ada.

Hasil pengujian dari algoritma yang dikembangkan ini tidak lebih bagus daripada algoritma bilinear tetapi masih lebih bagus daripada algoritma replikasi pixel. Hal ini disebabkan karena perbandingan jumlah pixel yang menentukan batas objek jauh lebih kecil daripada jumlah pixel yang tidak.

Kata kunci— citra, algoritma pengubahan ukuran citra, gradien warna.

1. LATAR BELAKANG

Dalam kehidupan sehari-hari manusia tidak pernah lepas dari teknologi informasi. Seiring berjalannya waktu kebutuhan manusia akan teknologi informasi pun semakin meningkat. Salah satu teknologi informasi yang masih berkembang pada saat ini adalah teknologi pengolahan

citra (*image processing*). Teknologi ini telah digunakan seperti pada kamera digital, laptop, televisi, dan lain-lain.

Format *JPEG* adalah format yang sering sekali dipakai untuk pengkompresian citra. Keuntungan dari format ini adalah format ini berbasis *bitmap* sehingga bagus digunakan untuk citra yang berupa panorama maupun foto dimana detail dari citra sangatlah penting dan hal tersebut akan sukar dilakukan jika menggunakan format png yang menggunakan *vector*. Oleh karena itu kamera-kamera yang biasanya digunakan menggunakan format jpg dalam pengompresiannya selain format *raw* yang begitu besar.

Namun format pengompresian yang sering dipakai ini tidaklah *lossless* (pasti ada yang dihilangkan). Sehingga sangat rentan untuk dilakukan proses *resize* tidak seperti format lain seperti png yang menyimpan citra dalam bentuk *vector* sehingga dapat dengan mudah diperbesar. Tujuan proses perbesaran ini adalah untuk mendapatkan citra dengan resolusi lebih tinggi dari citra sumber yang memiliki resolusi rendah. Jika citra *bitmap* diperbesar maka akan terdapat beberapa *pixel* tambahan yang tidak diketahui warnanya, oleh karena itu dibutuhkan algoritma pengubahan ukuran citra yang bagus untuk dapat mengestimasi warna dari *pixel-pixel* tambahan tadi.

Pada umumnya algoritma pengubahan ukuran gambar menggunakan input berupa citra RGB dan bertujuan untuk menghasilkan citra yang beresolusi lebih tinggi dari pada citra yang asli. Terdapat beberapa algoritma umum yang sering dipakai dalam pengubahan ukuran citra ini yaitu replikasi *pixel*, interpolasi bilinear, dan interpolasi *bicubic* [1]. Dari ketiga algoritma tersebut algoritma replikasi *pixel* adalah algoritma pengubahan ukuran citra yang paling banyak menyebabkan efek kotak-kotak pada citra hasil. Sedangkan algoritma interpolasi bilinear dan interpolasi *bicubic* memanfaatkan *first-order spline* dan *second-order spline*.

Adapun penelitian terkait yang dilakukan oleh Tan [5] dari Southern University di New Orleans. Tan memanfaatkan analisis gradien warna di antara *pixel-pixel* yang diketahui. Tan berpendapat bahwa nilai *pixel-pixel* tambahan akibat efek perbesaran citra yang dekat dengan *pixel* yang memiliki gradien rendah seharusnya berubah perlahan-lahan. Sedangkan untuk yang dekat dengan *pixel* yang bergradien tinggi akan lebih cepat perubahannya. Penelitian ini menghasilkan citra perbesaran dengan batas citra dalam suatu citra menjadi lebih jelas. Hal ini menyebabkan tulisan akan nampak lebih jelas jika diperbesar dengan algoritma ini dibandingkan dengan algoritma interpolasi bilinear dan memiliki akurasi yang kurang lebih sama dengan algoritma interpolasi *bicubic*, tetapi algoritma ini berjalan lebih cepat dari algoritma interpolasi *bicubic*.

Pada tugas akhir ini dibahas tentang algoritma baru untuk perubahan ukuran citra pada citra *bitmap* yang diharapkan bisa memperbaiki kelemahan citra *bitmap* dalam hal perubahan ukuran citra sehingga dapat menghasilkan sebuah citra hasil yang memiliki resolusi yang lebih besar dari citra asli yang menyerupai citra aslinya.

2. DASAR TEORI

Dalam dasar teori akan dibahas beberapa teori yang dipakai dalam makalah ini, yaitu tentang citra, pengolahan citra, algoritma pengolahan citra, serta penelitian terkait yang diacu dari makalah ini.

2.1 CITRA

Citra dapat direpresentasikan menjadi fungsi dua dimensi $f(x,y)$ yang setiap titiknya terdapat amplitudo intensitas dari citra tersebut di titik tersebut [2]. Citra yang nilai koordinat dari setiap titiknya dan amplitudo pada titik tersebut terbatas serta diskrit disebut citra digital [2]. Terdapat beberapa jenis citra digital, yaitu citra biner, citra *grayscale*, dan citra warna [4].

2.1.1 CITRA BINER

Citra yang terdiri atas dua warna, yaitu hitam dan putih sehingga hanya dibutuhkan satu bit untuk menyimpan warna. Nilai bit tersebut adalah nol atau satu, nilai nol untuk hitam dan satu untuk putih.

2.1.2 CITRA GRAYSCALE (SKALA KEABUAN)

Citra yang terdiri atas warna bergradasi dari hitam ke putih. Jumlah gradasi ini tergantung dari jumlah bit yang disediakan, sehingga jika terdiri atas dua bit maka terdapat dua pangkat dua atau empat gradasi warna abu-abu, jika tiga bit maka terdapat delapan gradasi warna.

2.1.3 CITRA WARNA (TRUE COLOR)

Citra yang setiap *pixel*-nya berisi informasi kombinasi dari tiga buah warna dasar (RGB = Red Green

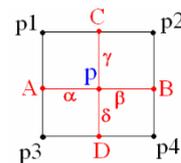
Blue). Ukuran penyimpanan warna dasar ini sebesar 8 bit atau 1 *byte*, sehingga jumlah warna yang tersedia sebanyak 255 tingkat warna untuk masing-masing warna dasar. Dengan 8 bit untuk masing-masing warna dasar maka terdapat 255x255x255 kombinasi warna yang dapat dihasilkan. Karena menyerupai dengan citra aslinya maka ada yang menyebutnya *true color*.

2.2 PENGOLAHAN CITRA

Pengolahan citra adalah segala proses yang dilakukan dengan masukan berupa citra. Keluaran dari pengolahan citra ini bisa berupa citra bisa juga berupa parameter. Pada umumnya pengolahan citra bertujuan untuk memperbaiki kualitas citra [6]. Beberapa contoh pengolahan citra yang sudah ada seperti akuisisi citra, filtering, konversi citra, *scaling*, dan lain-lain. *Image scaling* merupakan proses untuk mendapatkan citra yang beresolusi lebih tinggi dari citra sumber.

2.3 ALGORITMA PENGUBAHAN UKURAN CITRA

Terdapat beberapa algoritma perubahan ukuran citra yang biasa dipakai, yaitu replikasi *pixel*, interpolasi bilinear, dan interpolasi *bicubic*. Inti algoritma bilinear adalah menganggap warna diantara dua *pixel* yang diketahui adalah linier (berubah secara beraturan). Pada model ini warna suatu *pixel* ditentukan oleh empat buah *pixel* yang mengelilinginya, untuk lebih jelasnya lihat Gambar 1 [5].



Gambar 1 Interpolasi bilinear [5]

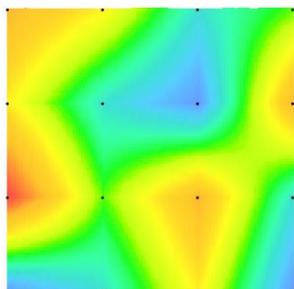
Titik p_1 , p_2 , p_3 , dan p_4 pada Gambar 1 adalah *pixel* yang diketahui warnanya sedangkan p adalah *pixel* yang ingin dicari warnanya dengan pendekatan interpolasi bilinear. Diambil titik A, B, C, dan D yang merupakan proyeksi titik p ke garis p_1-p_3 , p_2-p_4 , p_1-p_2 , dan p_3-p_4 . Didefinisikan α , β , γ , dan δ sebagai berikut 2.1.

$$\alpha = \frac{|AP|}{|AB|}, \quad \beta = \frac{|BP|}{|AB|}, \quad \gamma = \frac{|CP|}{|CD|}, \quad \delta = \frac{|DP|}{|CD|} \quad (2.1)$$

$|AB|$ merupakan panjang AB, hal yang sama juga berlaku terhadap $|AP|$, dan lain-lain. Jarak antara dua buah *pixel* terdekat dianggap satu satuan. Dalam hal ini juga berlaku $\alpha + \beta = \gamma + \delta = 1$. Warna dari titik p akan ditentukan dari rumus berikut 2.2, dengan $warna(x)$ berarti nilai warna pada *pixel* x .

$$\text{warna}(p) = \beta * \delta * \text{warna}(p1) + \alpha * \delta * \text{warna}(p2) + \beta * \gamma * \text{warna}(p3) + \alpha * \gamma * \text{warna}(p4) \quad (2.2)$$

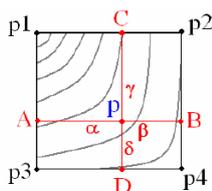
Jika sebuah citra berukuran $W \times H$ diperbesar sebesar n kali, maka ukuran citra tersebut akan menjadi $n*W \times n*H$. Perbesaran ini akan menyebabkan posisi *pixel* yang diketahui berubah sehingga dalam algoritma ini perlu dicari posisi *pixel* yang diketahui yang baru sehingga dapat mengestimasi warna bagi *pixel-pixel* yang baru. Gambar 2 adalah contoh citra hasil perbesaran interpolasi bilinear.



Gambar 2 Contoh citra hasil interpolasi bilinear [3]

2.4 PENELITIAN TERKAIT

Terdapat penelitian yang dilakukan oleh Delin Tan, dkk. Ide mereka adalah jika $\text{gradien}(p1) < \text{gradien}(p4)$ maka seharusnya warna pada *pixel-pixel* di dekat titik $p1$ berubah lebih lambat daripada *pixel-pixel* di dekat $p4$ (Gambar 3). Sehingga warna pada p pada algoritma ini akan lebih mirip ke $p1$ daripada warna p pada algoritma bilinear.



Gambar 3 Model analisis gradien [5]

Proses implementasi algoritma ini adalah pertama baca sebuah citra ke bitmap, kemudian simpan nilai RGB ke dalam tiga buah array dua dimensi. Setelah itu hitung luminan / intensitas dari tiap *pixel* pada rumus 2.3.

$$\text{Luminance} = (\text{nilai merah} + \text{nilai hijau} + \text{nilai biru})/3 \quad (2.3)$$

Setelah itu hitung GX dan GY yang merupakan gradien X dan Y pada masing-masing *pixel* seperti rumus 2.4 dan 2.5 dengan $\text{lum}(x,y)$ adalah nilai *luminance* pada titik x,y .

$$GX(x, y) = 2 * [\text{lum}(x+1, y) - \text{lum}(x-1, y)] + [\text{lum}(x+1, y+1) - \text{lum}(x-1, y+1)] + [\text{lum}(x+1, y-1) - \text{lum}(x-1, y-1)]; \quad (2.4)$$

$$GY(x, y) = 2 * [\text{lum}(x, y+1) - \text{lum}(x, y-1)] + [\text{lum}(x-1, y+1) - \text{lum}(x-1, y-1)] + [\text{lum}(x+1, y+1) - \text{lum}(x+1, y-1)]; \quad (2.5)$$

Dari GX dan GY pada tiap-tiap *pixel* dicari gradien dari tiap *pixel* seperti rumus 2.6.

$$\text{Gradient}(x, y) = \sqrt{GX(x, y)^2 + GY(x, y)^2} / 8 \quad (2.6)$$

Setelah itu dicari kombinasi $\alpha, \beta, \gamma,$ dan δ yang sesuai dengan $\text{Gradien}(x,y)$ untuk mencari nilai warna *pixel* pada posisi x,y .

3. ANALISIS ALGORITMA

Pada bagian ini akan dibahas mengenai perancangan algoritma yang diajukan, mulai dari perhitungan letak *pixel* dan algoritma perubahan ukuran itu sendiri.

3.1 ANALISIS LETAK PIXEL

Letak *pixel* pada citra asal dan citra yang telah diubah ukurannya akan berbeda, maka dari itu digunakan rumus 3.1 dalam perhitungannya.

$$\text{posisi baru} = \frac{\text{posisi lama}}{\text{posisi maksimal lama}} \times \text{posisi maksimal baru} \quad (3.1)$$

3.2 ANALISIS ALGORITMA PERUBAHAN UKURAN

Dasar dari algoritma yang dikembangkan ini adalah algoritma bilinear. Untuk rumus dari algoritma bilinear dapat diubah seperti pada rumus 3.2.

$$\alpha = \frac{f(AP)}{f(AB)}, \beta = 1 - \alpha, \gamma = \frac{f(CP)}{f(CD)}, \delta = 1 - \gamma \quad (3.2)$$

Dalam algoritma bilinear digunakan $f(x)=x$. Maka dari itu dicari sebuah $f(x)$ yang dapat menerapkan prinsip analisis gradien yang dikembangkan oleh Tan dan kawan-kawan. Setelah melakukan beberapa simulasi rumus $f(x)$ maka dipilihlah $f(x,p)=x^p$ dimana p adalah nilai pangkat yang dapat diatur. Kemudian dengan menggunakan prinsip Tan dan kawan-kawan dicari nilai warna dari *pixel* yang belum diketahui.

4. PENGUJIAN

Pada bagian ini akan dibahas pengujian yang dilakukan dalam menguji algoritma ini. Pengujian ini dilakukan dengan nilai p adalah 2 dan algoritma pembandingnya adalah algoritma replikasi *pixel* dan algoritma bilinear.

4.1 PENGUJIAN BATAS ANTAR OBJEK

Pengujian ini digunakan citra logo ITB berukuran 128 x 128 *pixel* yang diperbesar sebesar 8x. Hasil pengujian ini dapat dilihat pada Tabel 1.

Tabel 1 Pengujian batas citra

Replikasi <i>pixel</i>	
Bilinear	
Polinomial pangkat dua	

akan dibandingkan dengan citra awal. Perbandingan yang dilakukan dengan menggunakan PSNR (*Peak Signal-to-Noise*) dan CC (*Cross-Correlation*) dimana semakin besar nilainya akan semakin bagus.



Gambar 4 Citra uji

Tabel 2 adalah tabel nilai PSNR dari keenam belas citra uji untuk ketiga algoritma yang digunakan.

Tabel 2 Nilai PSNR citra uji

Nomor Citra	Replikasi <i>Pixel</i>	Bilinear	Polinomial pangkat dua
1	25.0501	25.6935	25.0374
2	24.6549	25.8081	25.1032
3	27.1794	27.5985	27.0148
4	19.7668	20.9084	20.3143
5	31.6018	31.6896	31.0435
6	18.5327	18.7619	19.0287
7	25.0950	25.5750	25.0517
8	20.4536	20.5190	19.9826
9	26.0579	26.0370	25.5081
10	36.2349	39.4960	38.8289
11	32.2258	31.2913	30.7689
12	22.9166	23.5276	22.9929
13	19.6953	20.2742	19.7772
14	22.2402	22.5908	22.1045
15	20.2944	21.41	20.7815
16	24.5196	25.2875	24.667
Rata-rata	24.7824	25.4043	24.8753

4.2 PENGUJIAN DENGAN CITRA UJI

Pengujian ini menggunakan enam belas citra uji yang terdiri atas beberapa citra seperti citra *true color*, citra *grayscale*, citra teks, citra logo, citra abstrak, citra pemandangan, dan lain-lain. Citra-citra tersebut (pada Gambar 4) diperkecil sebesar setengah kali ukuran semula kemudian diperbesar lagi sebesar dua kali dan hasilnya



Tabel 3 adalah tabel nilai CC dari keenam belas citra uji untuk ketiga algoritma yang digunakan.

Tabel 3 Nilai CC citra uji

Nomor Citra	Replikasi Pixel	Bilinear	Polinomial pangkat dua
1	0.9772	0.9803	0.9773
2	0.9492	0.9605	0.9540
3	0.9743	0.9764	0.9735
4	0.8135	0.8477	0.8281
5	0.9954	0.9955	0.9948
6	0.9557	0.9573	0.9602
7	0.9728	0.9757	0.9732
8	0.9184	0.9173	0.9075
9	0.9856	0.9855	0.9837
10	0.9898	0.9952	0.9947
11	0.9944	0.9932	0.9925
12	0.9532	0.9593	0.9547
13	0.8910	0.9026	0.8923
14	0.9342	0.9378	0.9310
15	0.9147	0.9322	0.9226
16	0.9558	0.9625	0.9572
Rata-rata	0.9484	0.9549	0.9498

5. ANALISIS HASIL PENGUJIAN

Pada bagian ini akan dibahas analisis dari hasil pengujian yang telah dibahas pada bagian sebelumnya

5.1 ANALISIS BATAS ANTAR OBJEK

Algoritma replikasi *pixel* terlihat kotak-kotak karena efek *blocky* yang terlalu besar. Untuk algoritma bilinear dan algoritma baru ini terlihat lingkaran hitam yang jelas, namun pada algoritma bilinear terlihat lebih pudar karena pembagian warna yang merata dari hitam ke putih. Pada algoritma yang berdasarkan analisis gradien dengan pendekatan polinomial ini lingkaran hitam tersebut terlihat lebih jelas dikarenakan adanya efek *blocky* yang kecil pada *pixel-pixel* yang berada pada batas dari citra lingkaran hitam tersebut.

5.2 ANALISIS CITRA UJI

Setelah melihat data nilai PSNR dan CC dari keenam belas citra uji maka dapat diambil kesimpulan bahwa algoritma baru ini masih berada diantara algoritma replikasi *pixel* dan algoritma bilinear. Algoritma baru ini paling bagus bila digunakan untuk memperbesar teks (lihat hasil PSNR dan CC citra uji keenam). Hal ini juga berlaku dengan percobaan lain yang dilakukan terhadap citra teks.

6. KESIMPULAN

Dari pengujian serta analisis yang telah dilakukan sebelumnya dapat diambil beberapa kesimpulan:

1. Algoritma yang dikembangkan ini dapat mempertajam batas antar objek dalam suatu citra.
2. Algoritma ini masih berada diantara bilinear dan replikasi *pixel* dengan nilai PSNR dan CC bilinear yang paling tinggi.
3. Algoritma ini paling cocok digunakan untuk citra berjenis teks.

7. REFERENSI

- [1] Gao, R., Song, J.-P., & Tai, X.-C. (2009). *Image Zooming Zooming Algorithm Based on Partial Differential Equations Technique*. *International Journal of Numerical Analysis and Modelin*, 284-292.
- [2] Gonzalez, R. C., & Woods, R. E. (2001). *Digital Image Processing Second Edition*. New Jersey: Prentice-Hall.
- [3] Lancaster, D. (2007). *A Review of Some Image Pixel Interpolation Algorithm*. Retrieved 1 7, 2013, from The Guru's Lair: <http://www.tinaja.com/glib/pixintpl.pdf>
- [4] Sirait, H. (2009). *Citra Digital*. Retrieved 1 6, 2013, from HASANUDDIN SIRAIT: <http://siraith.files.wordpress.com/2009/02/presentation1.pdf>
- [5] Tan, D. (2000). *Image Zooming Algorithm Based on Gradient Analyzing Model*. Retrieved 12 17, 2012, from <http://www.sunocas.com/tan/papers/Image%20Zooming%20Algorithm%20Based%20on%20Color%20Data%20Gradient%20Analyzing.pdf>
- [6] Wibowo, B. (2012). *Teks, Gambar dan Grafik*.