



Simulasi Penyembunyian *Error* pada Citra Menggunakan Metode *Multi Directional Interpolation* (MDI)

Riko Arlando Saragih dan Hanapi Gunawan

Jurusan Teknik Elektro, Universitas Kristen Maranatha, Bandung
Jl. Suria Sumantri 65, Bandung 40164, Indonesia
riko_saragih@yahoo.com, hanapigunawan@gmail.com

Abstrak: Pada prakteknya saluran komunikasi tidak terbebas dari gangguan. Gangguan pada saluran komunikasi dapat menyebabkan data citra yang diterima menjadi rusak atau ada yang hilang. Hal ini menyebabkan kualitas citra menjadi berkurang. Salah satu cara untuk mengatasi *error* pada citra adalah dengan *error concealment* atau penyembunyian *error*. Tulisan ini menjelaskan suatu metode penyembunyian *error* dengan menggunakan informasi spasial di sekitar blok yang rusak, sehingga sebuah objek pada citra yang rusak dapat dikenali. Metode penyembunyian *error* yang digunakan dalam tulisan ini adalah metode Multi Directional Interpolation (MDI). Metode *Multi Directional Interpolation* (MDI) ini menyembunyikan *error* dengan menggunakan informasi spasial dari blok tetangga. Interpolasi dilakukan berdasarkan pemilihan arah tepi yang sesuai, yang dalam tulisan ini dipilih maksimal tiga arah tepi. Berdasarkan hasil simulasi untuk pengujian metode ini, diperoleh bahwa untuk blok *error* ukuran 8 x 8 piksel dapat dicapai rata-rata kenaikan PSNR (*Peak Signal Noise to Ratio*) sebesar 13,52 dB.

Kata kunci: Penyembunyian *Error*, Metode MDI.

Abstract: In practice, communication channels are not noise-free. Noise in the communication channels can cause damages or loss in image data, and ultimately it will reduce the quality of the images. One way to ameliorate the error damage on images is by error concealment. This article describes a method to do the error concealment that using spatial information around the damaged block, so that an object in damages images could be recognizable. Error concealment method that was used in this article is Multi Directional Interpolation (MDI). This MDI method conceals the errors by using the spatial information from the neighborhood. The interpolation was done by choosing the appropriate directional edges, which was being chosen maximal three in this paper. Based on the simulation result for this method, it can be achieved that for block error 8 x 8 pixel, the average PSNR (*Peak Signal Noise to Ratio*) increases about 13.52dB.

Keywords: Error Concealment, MDI Method

I. PENDAHULUAN

Untuk mentransmisikan sebuah citra, citra terbagi-bagi menjadi beberapa paket data. *Noise* pada jaringan menyebabkan paket data yang dikirim mengalami penundaan atau *error*, sehingga data yang diterima tidak lengkap. Jika ada citra yang rusak pada saat pengiriman, maka perlu dilakukan pengiriman ulang (*retransmission*). Pengiriman ulang yang dilakukan akan membutuhkan waktu. Untuk mengatasinya, dekoder dapat mengganti data informasi yang hilang dengan bit-bit yang sudah ada dengan teknik penyembunyian *error*.

Ada beberapa metode penyembunyian *error* yang digunakan seperti Pixel Domain Interpolation (PDI), Multi Directional Interpolation (MDI), Projection Onto Convex Sets (POCS), Multiple description coding (MDC) dan Maximally Smooth Recovery (MSR). Tulisan ini akan menjelaskan proses penyembunyian *error* pada citra yang rusak dengan menggunakan metode Multi-Directional Interpolation (MDI) serta menyajikan hasil simulasi dan analisisnya.

II. TEORI TENTANG CITRA

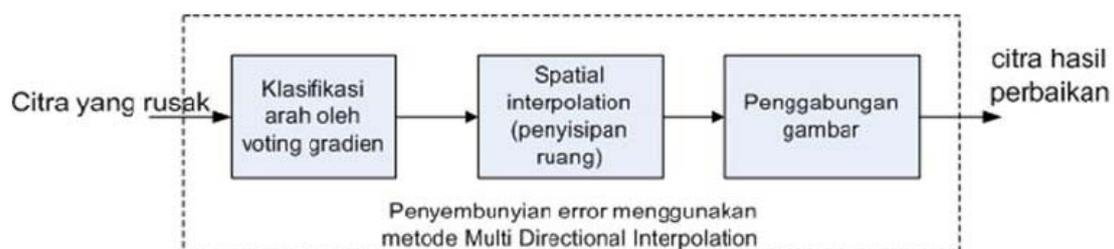
Citra (*image*) adalah gambar pada bidang dwimarta (2 dimensi). Dari sudut pandang matematis, citra merupakan fungsi kontinu dari intensitas cahaya pada bidang 2 dimensi. Dua informasi mendasar pada citra adalah *luminance*, yaitu tingkat terang-gelap dan *chrominance* yang memberikan informasi warna. Citra ada 2 macam, yaitu citra kontinu dan citra diskrit. Citra kontinu dihasilkan dari sistem optik yang menerima sinyal analog. Citra diskrit dihasilkan melalui proses digitalisasi dari citra kontinu, citra diskrit sering disebut citra digital.^[1]

Citra yang akan dikirim melewati saluran transmisi dibagi menjadi beberapa paket data yang lebih kecil. *Noise* pada saluran transmisi mengakibatkan beberapa paket data rusak. Pada paket data yang rusak, sulit untuk mengetahui berapa banyak data dalam paket yang dapat digunakan.

Untuk lebih mudahnya, paket yang rusak dianggap sebagai paket yang hilang. Paket data yang hilang pada citra digambarkan sebagai blok yang hilang atau blok yang rusak.^[2]

Jika terjadi *error* pada saat transmisi atau pada saat pengompresan data, maka akan dilakukan pengiriman ulang. Pengiriman ulang pada citra yang berukuran besar akan membutuhkan waktu. Untuk mengatasi hal itu, salah satu caranya adalah dengan penyembunyian *error*. Penyembunyian *error* yang dimaksud adalah menutup blok yang rusak pada citra dengan sebuah blok yang nilai di dalamnya diambil dari nilai-nilai di sekitar blok yang rusak tersebut.

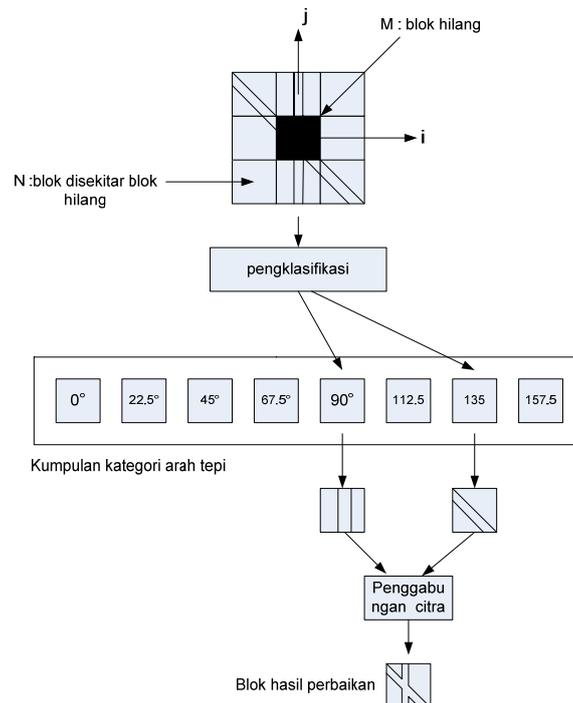
III. MULTI DIRECTIONAL INTERPOLATION



Gambar 1. Diagram blok proses penyembunyian *error*

Metode MDI (Multi Directional Interpolation) pada citra adalah salah satu proses perbaikan pada citra yang rusak dengan cara menutup blok yang rusak dengan menggunakan interpolasi spasial. Secara garis besar, proses penyembunyian error terdiri dari tiga proses seperti yang ditunjukkan dalam Gambar 1.

Metode ini menggunakan operator Sobel untuk mendeteksi informasi tepi. Setelah itu menggunakan klasifikasi voting *gradient* untuk mengambil dua atau tiga nilai tepi yang terkuat di sekitar blok yang rusak dan menampilkan interpolasi spasial dari berbagai arah tepi yang kuat untuk menutup blok yang rusak. Ciri-ciri yang kuat dari berbagai arah interpolasi diekstrak dan bersama-sama digabungkan dalam satu blok dengan menggunakan operator MAX (nilai piksel yang terbesar), operator MIN (nilai piksel yang terkecil), dan AVG (nilai piksel rata-rata). Secara garis besar, proses penyembunyian error ditunjukkan oleh Gambar 2.



Gambar 2. Proses penyembunyian error dengan MDI

IV. PENDETEKSIAN TEPI

Tepi adalah perubahan nilai intensitas derajat keabuan yang mendadak (besar) dalam jarak yang singkat. Tepi biasanya terdapat pada batas antara dua daerah berbeda pada suatu citra. Tepi dapat diorientasikan dengan suatu arah, dan arah ini berbeda-beda bergantung pada perubahan intensitas. Pendeteksian tepi merupakan langkah pertama untuk melingkupi informasi di dalam citra. Tepi mencirikan batas-batas objek dan oleh karena itu, tepi berguna untuk proses segmentasi dan identifikasi objek di dalam citra. Tujuan pendeteksian tepi adalah untuk meningkatkan penampakan garis batas suatu daerah atau objek dalam citra .

Untuk melakukan pendeteksian tepi, dapat digunakan operator gradien pertama dan operator Sobel. Pendeteksian tepi dengan menggunakan operator gradien pertama dapat dijelaskan sebagai berikut :^[1]

Perubahan intensitas yang besar dalam jarak yang singkat dipandang sebagai fungsi

yang memiliki kemiringan yang besar. Kemiringan fungsi biasanya dilakukan dengan menghitung turunan pertama (gradien). Karena citra $f(x,y)$ adalah fungsi dwimatra (dua dimensi) dalam bentuk diskrit, maka turunan pertamanya adalah secara parsial, masing-masing dalam arah x dan arah y sebagai berikut:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} \quad (1)$$

yang dalam hal ini :

$$G_x = \frac{\partial f(x, y)}{\partial x} = \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x} \quad (2)$$

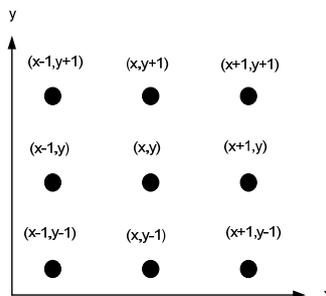
$$G_y = \frac{\partial f(x, y)}{\partial y} = \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y} \quad (3)$$

Apabila $\Delta x = \Delta y = 1$, persamaan turunan pertama menjadi:

$$G_x = \frac{\partial f(x, y)}{\partial x} = f(x + 1, y) - f(x, y) \quad (4)$$

$$G_y = \frac{\partial f(x, y)}{\partial y} = f(x, y + 1) - f(x, y) \quad (5)$$

Titik-titik yang digunakan dalam perhitungan turunan pertama diperlihatkan dalam Gambar 3.



Gambar 3. Titik-titik yang digunakan dalam perhitungan gradien

Kedua turunan pada Persamaan 4 dan Persamaan 5 dapat dipandang sebagai dua buah mask konvolusi sebagai berikut :

$$G_x = [-1 \ 1] \quad \text{dan} \quad G_y = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Berdasarkan konvolusi dengan kedua *mask* tersebut, magnitudo dan arah dari gradien dapat dihitung untuk setiap piksel :

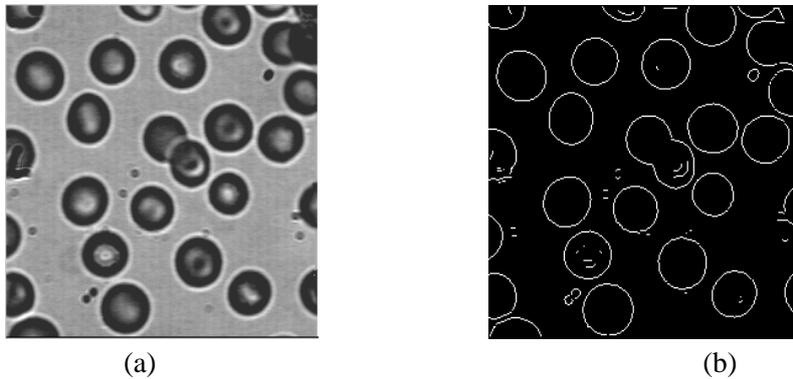
$$G[f(x, y)] = \sqrt{G_x^2 + G_y^2} \quad (6)$$

$$\alpha(x, y) = \tan^{-1} \frac{G_x}{G_y} \quad (7)$$

Keputusan apakah suatu piksel merupakan tepi atau bukan, dinyatakan dengan operasi pengambangan (*threshold*) berikut :

$$G(x, y) = \begin{cases} 1, & \text{jika } G[f(x, y)] \geq T \\ 0, & \text{lainnya} \end{cases} \quad (8)$$

T adalah nilai ambang batas, piksel tepi dinyatakan putih sedangkan piksel bukan tepi dinyatakan hitam. Gambar 4 memperlihatkan contoh hasil pendeteksian tepi dengan menggunakan operator gradien pertama.



Gambar 4. (a) original image, (b) tepi(edge) image

Di dalam melakukan pendeteksian tepi, prinsip kerja dari Operator *Sobel* ini hampir sama dengan Operator Gradien Pertama.^[3] Tinjau pengaturan piksel di sekitar piksel(x,y) sebagai berikut :

$$\begin{bmatrix} a_0 & a_1 & a_2 \\ a_7 & (x, y) & a_3 \\ a_6 & a_5 & a_4 \end{bmatrix}$$

Dalam hal ini turunan parsial dihitung dengan cara sebagai berikut :

$$S_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6) \quad (9)$$

$$S_y = (a_0 + ca_1 + a_2) - (a_6 + ca_5 + a_4) \quad (10)$$

Dengan konstanta $c=2$, dalam bentuk mask konvolusi dapat dinyatakan sebagai berikut :

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{dan} \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Sementara arah tepi dihitung dengan persamaan :

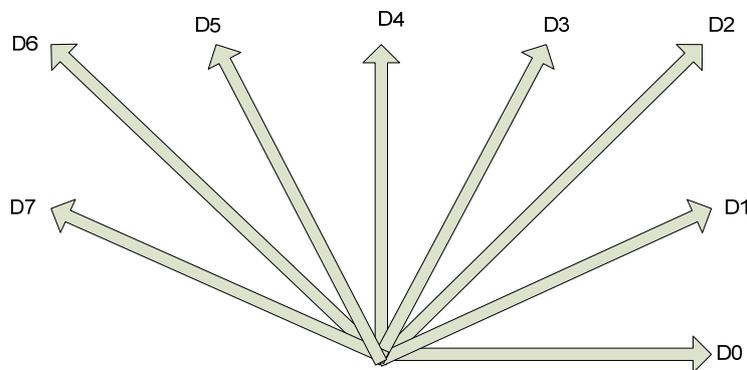
$$\alpha(x, y) = \tan^{-1} \left(\frac{S_y}{S_x} \right) \quad (11)$$

Magnitude dari gradien dihitung dengan persamaan :

$$M = \sqrt{S_x^2 + S_y^2} \quad (12)$$

V. KLASIFIKASI OLEH VOTING GRADIEN

Pada klasifikasi *voting gradien* arah tepi dibagi menjadi 8 kategori, masing-masing arah merupakan kelipatan $22,5^\circ$. Untuk masing-masing arah tersebut terdapat *counter* (pencacah) yang dimulai dari D0 sampai D7^[2], seperti yang ditunjukkan dalam Gambar 5.



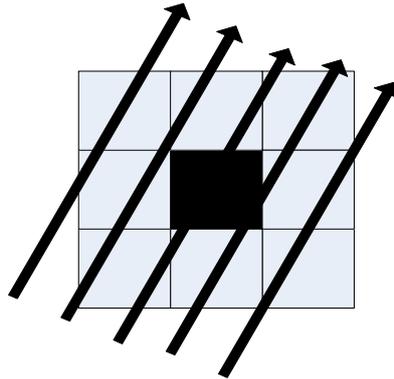
Gambar 5. Arah Tepi

Untuk masing-masing counter diwakili oleh sudut yang berbeda, yaitu D0 (0°), D1 ($22,5^\circ$), D2 (45°), D3 ($67,5^\circ$), D4 (90°), D5 ($112,5^\circ$), D6 (135°), dan D7 ($157,5^\circ$). Setelah semua nilai magnitude dan arahnya dihitung, tiga nilai magnitude terbesar beserta arahnya diambil untuk dijadikan acuan pada tahap selanjutnya, yaitu interpolasi spasial.

VI. INTERPOLASI SPASIAL

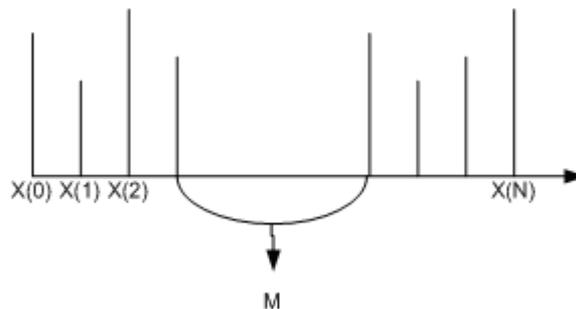
Spasial adalah suatu daerah pada gambar yang mengandung informasi berupa intensitas piksel. Sedangkan interpolasi didefinisikan sebagai suatu cara memperkirakan nilai fungsi di antara poin data yang telah diketahui. Proses interpolasi spasial pada *error concealment* adalah merekonstruksi piksel-piksel yang hilang menggunakan informasi redundansi spasial dari piksel-piksel di sekitarnya. Proses interpolasi spasial dibutuhkan untuk menyembunyikan efek dari blok yang rusak.

Interpolasi spasial dipakai untuk setiap arah yang telah dihitung dari pendeteksi tepi. Sebagai acuannya, satu paket interpolasi satu dimensi dipakai pada arah tepi yang telah ditentukan.^[2] Hal ini diperlihatkan pada Gambar 6.



Gambar 6. Satu paket interpolasi satu dimensi

Interpolasi satu dimensi yang dilakukan akan menghasilkan sebuah array satu dimensi yang berisi nilai-nilai piksel yang dilewati oleh arah tepi pada daerah di sekitar blok yang rusak. Array yang didapat dari interpolasi satu dimensi ditunjukkan pada Gambar 7 berikut.^[2]



Gambar 7. Array satu dimensi

Dengan M adalah nilai-nilai piksel yang hilang, $x(N)$ adalah nilai-nilai piksel yang didapat dari blok di sekitar blok yang rusak.

VII. INTERPOLASI SPASIAL

Tujuan dari penggabungan citra adalah untuk mengekstrak ciri-ciri karakter yang kuat dari 2 citra atau lebih dan menggabungkannya ke dalam satu citra. Pertama-tama harus dapat mengklasifikasi nilai-nilai menjadi *background* dan *foreground*. Hal ini dapat dilaksanakan dengan menghitung histogram dari masing-masing citra, varian dan rata-rata nilai pikselnya. Piksel yang mendekati nilai rata-rata akan dijadikan *background* (BG), sedangkan piksel dengan nilai yang jauh dari nilai rata-rata dijadikan *foregorund*. Dengan menggunakan ambang batas (*threshold*) tertentu, piksel yang lebih besar dari rata-rata dan lebih besar dari nilai *threshold*nya dijadikan *bright foreground* (BF) sedangkan piksel yang lebih kecil dari rata-rata dan lebih kecil dari nilai *threshold*nya dijadikan *dark foreground* (DF). Untuk mengekstrak ciri-ciri *foreground* yang terang, dipilih nilai piksel yang maksimum dari dua citra dengan menggunakan operator MAX. Demikian juga untuk mengekstrak *foregorund* yang gelap, operator MIN digunakan pada piksel *foreground* tersebut. Untuk mengkombinasikan ciri-ciri *background* dari dua citra maka, nilai piksel dari dua citra tersebut dirata-ratakan^[2]. Gambar 8 memperlihatkan operator yang digunakan.

CITRA 2

		Bright foreground	background	Dark foreground
CITRA 1	Bright foreground	MAX	MAX	AVG
	background	MAX	AVG	MIN
	Dark foreground	AVG	MIN	MIN

Gambar 8. Operator MAX, MIN, AVG

VIII. KUALITAS CITRA

Kualitas citra secara objektif ditentukan oleh PSNR (*Peak Signal to Noise Ratio*). Dalam tulisan ini, PSNR digunakan untuk menganalisa metode penyembunyian error yang dilakukan. Rumus PSNR dinyatakan sebagai berikut :

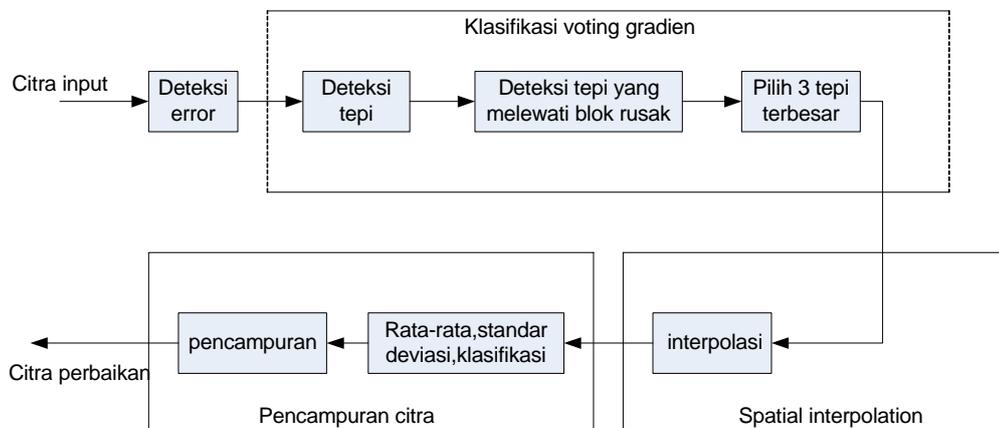
$$PSNR = 10 \log \left(\frac{255^2}{MSE} \right) \tag{13}$$

Dengan MSE (*Mean Squared Error*) dinyatakan dengan :

$$MSE = \frac{1}{MN} \sum_{j=1}^M \sum_{i=1}^N [x(i, j) - x'(i, j)]^2 \tag{14}$$

Dengan $x(i, j)$ merupakan nilai piksel atau derajat keabuan pada koordinat (i, j) pada citra asli, sedangkan $x'(i, j)$ menyatakan nilai piksel atau derajat keabuan pada koordinat yang sama pada citra hasil rekonstruksi. MSE menunjukkan selisih (perbedaan) nilai piksel atau derajat keabuan antara citra asli dengan citra hasil rekonstruksi.

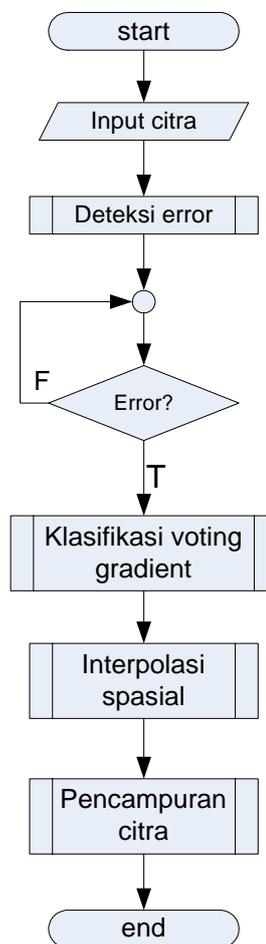
IX. DIAGRAM BLOK SISTEM PENYEMBUNYIAN ERROR SECARA LENGKAP



Gambar 9. Diagram Blok Sistem Penyembunyian Error secara Lengkap

Sistem penyembunyian error yang dirancang terbagi ke dalam empat modul utama, dengan blok lengkap rancangan sistem seperti diperlihatkan dalam Gambar 9.

Dalam Gambar 9, citra input adalah citra yang sudah mengandung *error* (sengaja dirusak dalam penelitian ini). Citra ini diperoleh dari citra asli yang nilai pikselnya diset sama dengan nol untuk tiap blok yang berindeks genap. Proses deteksi *error* dilakukan untuk memeriksa blok citra mana yang mengandung *error* (seluruh piksel dalam blok tersebut bernilai nol). Pada blok “Klasifikasi voting gradien” dilakukan proses pemilihan jenis tepi yang menghasilkan nilai *error* yang terbesar. Proses penyembunyian *error* terutama dilakukan pada blok “Spatial interpolation”. Pada tahap akhir dari proses penyembunyian *error* dilakukan pencampuran (penggabungan) antara blok citra yang tidak mengalami *error* (nilai piksel dalam blok tersebut *tidak nol semuanya*) dengan blok citra yang sudah disembunyikan *error*-nya. Diagram alir dari proses penyembunyian *error* dalam penelitian dapat dilihat dalam Gambar 10.



Gambar 10. Diagram Alir Proses Penyembunyian *Error*

X. DATA PENGAMATAN DAN ANALISIS

Input citra yang digunakan pada simulasi ini adalah citra diam yang terlebih dulu dirusak, berupa blok-blok yang rusak. Pada simulasi ini karena metode yang digunakan memanfaatkan informasi dari blok yang tidak rusak di sekitarnya, maka blok-blok rusak

didistribusi menyebar secara teratur. Format citra yang digunakan .jpg dan .tif dengan mode warna *grayscale* [0-255]. Pada simulasi ini akan digunakan citra dengan ukuran 256 x 256 piksel dan 512 x 512 piksel. Citra yang rusak dibagi menjadi 8 x 8 blok, setelah dideteksi letak blok error, maka dilakukan pendeteksian tepi di sekitar blok yang rusak. Gambar 11 adalah contoh citra tanpa error.



Gambar 11. Citra tanpa error 'foreman.jpg (256 x 256 piksel)'

Untuk tahapan proses klasifikasi voting gradien, maka setelah nilai magnituda dan arah tepi di sekitar blok rusak diperoleh, diambil nilai magnituda terbesar yang melewati blok rusak, kemudian dikalikan dengan 0.75. Nilai tersebut digunakan sebagai batas untuk menentukan tiga magnituda terbesar disekitar blok yang rusak yang akan digunakan dalam menentukan arah interpolasi. Untuk setiap magnituda yang dipilih, terdapat arah magnitudanya yang telah diklasifikasikan menjadi 8 arah yaitu indeks 0 (0), indeks 1 (22,5), indeks 2(45), indeks 3(67,5), indeks 4(90), indeks 5(112,5), indeks 6(135), indeks 7(157,5). Maksimal tiga arah sudut yang diperoleh dari tiga nilai magnituda terbesar digunakan untuk acuan dalam proses interpolasi. Gambar 12 adalah contoh nilai magnituda dari blok di sekitar blok error yang terakhir. Untuk contoh nilai arah tepi pada blok di sekitar blok rusak yang terakhir ditunjukkan oleh Gambar 13.

	1	2	3	4	5	6	7	8	9	10
1	166.43	419.03	285.95	172.8	236.7	116.28	408.81	474.83	38.419	286.56
2	182.48	355.24	316.37	72.028	196.06	134.36	438.74	523.16	37.363	253.16
3	191.21	244.77	345.94	402.24	211.06	186.47	439.99	404.75	194.03	249.23
4	173.22	186.99	283.21	405.86	356.49	256.01	294.36	217.86	275.54	224.15
5	146.1	111.83	177.28	142.41	254.03	101.27	162.05	145.91	129.88	79.511
6	243.82	213.78	258.86	194.98	203.49	79.699	89.107	76.968	81.006	77.833

Gambar 12. Contoh Nilai Magnituda

	1	2	3	4	5	6	7	8	9	10
1	-4.1347	-0.68369	-10.681	52.997	31.877	49.185	12.574	3.3806	-51.34	9.035
2	-9.4623	2.0972	20.153	88.409	18.435	45.603	5.0998	1.4239	74.476	-7.49
3	8.7234	28.554	65.583	71.745	-88.643	-38.468	-13.405	7.8098	-81.703	-42.073
4	31.304	57.67	-77.976	65.392	57.603	-53.578	-15.972	43.512	-80.811	-68.816
5	-40.559	-54.462	23.962	-12.572	0.90222	9.0903	35.455	78.536	-80.248	83.501
6	-52.331	-79.216	43.748	-56.717	-27.195	17.526	9.0395	-24.567	-89.293	-69.702

Gambar 13. Contoh Nilai Arah Tepi

Untuk setiap arah tepi yang didapat, nilainya dibulatkan ke kelipatan $22,5^\circ$. Masing – masing arah tepi digantikan dengan indeks yang dimulai dari indeks 0 untuk sudut 0° sampai indeks 7 untuk sudut $157,5^\circ$. Gambar 14 memperlihatkan contoh dari indeks di sekitar blok rusak.

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	2	1	2	1	0	6	0
2	0	0	1	4	1	2	0	0	3	0
3	0	1	3	3	4	6	7	0	4	6
4	1	3	5	3	3	6	7	2	4	5
5	6	6	1	7	0	0	2	3	4	4
6	6	4	2	5	7	1	0	7	4	5

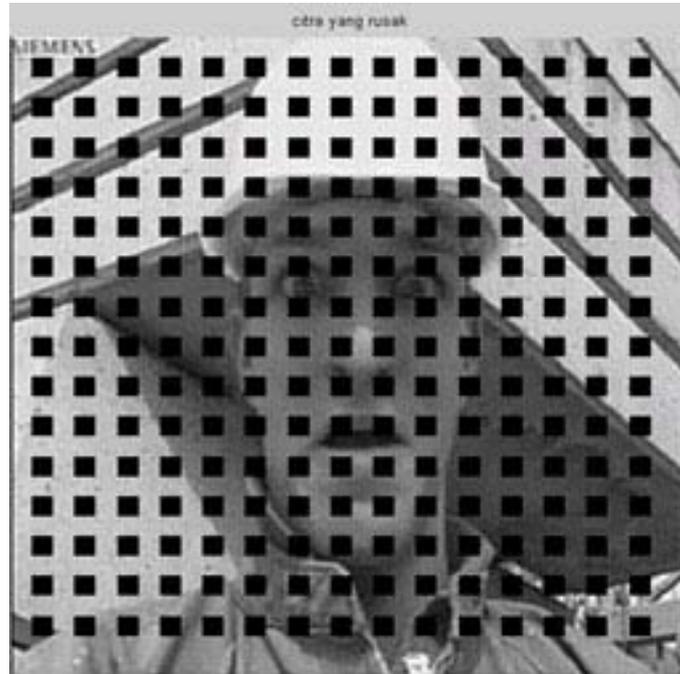
Gambar 14. Contoh Indeks

Setelah semua magnituda dan arah tepi didapat, diambil maksimal tiga arah tepi terbesar. Tabel 1 menampilkan contoh indeks arah tepi yang akan digunakan dalam proses interpolasi untuk setiap blok 8×8 piksel pada gambar 'foreman.jpg' dengan ukuran 256×256 piksel (hanya indeks arah tepi untuk posisi blok dari baris ke-2 dan untuk posisi blok dari kolom ke-2 sampai dengan kolom ke-30).

TABEL 1. INDEKS ARAH TEPI

Posisi blok pada baris ke- <i>i</i>	Posisi blok pada kolom ke- <i>i</i>	Indeks arah tepi
2	2	5,3, dan 2
2	4	5,3, dan 2
2	6	5,3, dan 2
2	8	5,3, dan 2
2	10	5,3, dan 2
2	12	6,3, dan 2
2	14	5,7, dan 2
2	16	7,2, dan 3
2	18	4,2, dan 3
2	20	2 dan 3
2	22	2 dan 3
2	24	2 dan 3
2	26	2 dan 3
2	28	0 dan 1
2	30	2 dan 1

Dari Tabel 1 dapat diperoleh nilai indeks arah tepi untuk setiap blok yang rusak. Sebagai contoh, untuk blok rusak yang pertama, yaitu pada posisi baris ke dua dan kolom kedua pada citra yang telah dibagi ke dalam blok 8 x 8 piksel, indeksnya 5,3,2. Jadi, pada blok error yang pertama, tiga magnituda terbesar mempunyai arah tepi $112,5^\circ$ (indeks 5), $67,5^\circ$ (indeks 3), dan 45° (indeks 2). Gambar 15 adalah contoh citra yang rusak.



Gambar 15. Citra yang rusak 'foreman.jpg' (256 x 256 piksel)

Pada tahapan proses interpolasi spasial, maka nilai indeks yang diperoleh dari klasifikasi *voting gradien* digunakan sebagai acuan untuk menarik garis lurus sesuai sudut arah tepinya pada daerah blok yang rusak. Nilai yang didapat dari penarikan garis digunakan untuk mencari nilai piksel pada blok yang rusak. Gambar 16 adalah contoh nilai hasil interpolasi untuk satu arah tepi.

147	143.73	146.96	144.5	144.52	138.55	138.48	137.4
152.9	149.29	149.29	132.75	135.79	131.77	128.3	119.87
144.74	147.36	145.12	144.86	138.8	138.53	137.72	138.02
150.03	149.56	132.83	135.34	131.55	128.44	120.35	113.79
148.16	145.69	145.41	139.52	139.24	138.48	139.04	135.54
150.37	132.87	135.33	131.71	128.78	120.77	114.13	112.29
146.78	146.78	141.49	141.39	140.44	140.94	137.46	136.62
133	135.96	132.55	129.68	121.48	114.5	112.1	103.77

Gambar 16. Contoh nilai hasil interpolasi

Selanjutnya pada tahapan proses pencampuran, maka untuk setiap satu titik di dalam blok yang rusak terdapat tiga nilai. Tiga nilai tersebut di klasifikasikan ke dalam tiga kategori

yaitu, *bright foreground* (BF), *dark foreground* (DF), *background* (BG). Untuk menentukan nilai yang akan menggantikan nilai piksel di dalam blok yang rusak, digunakan operator MAX yaitu mencari nilai yang terbesar, operator MIN yaitu mencari nilai yang terkecil, dan operator AVG yaitu mencari nilai rata-rata. Setelah blok di dalam citra dapat direkonstruksi, dilakukan penghitungan PSNR pada citra hasil penyembunyian *error*. Gambar 17 adalah contoh nilai hasil pencampuran.

156.65	143.73	150.99	147.51	135.1	148.51	147.86	145.7
145.2	149.29	149.29	132.75	148.87	131.77	145.94	146.95
150.6	151.76	152.24	149.16	148.26	146.13	147.1	147.34
145.98	149.56	132.83	135.34	146.33	128.44	147.57	113.79
153.05	145.69	145.41	139.52	147.52	147.84	144.51	144.22
147.17	132.87	135.33	131.71	148.24	120.77	114.13	112.29
149.91	154.5	141.49	141.39	145.86	145.22	135.14	131.84
133	135.96	132.55	129.68	121.48	114.5	112.1	103.77

Gambar 17. Contoh nilai hasil pencampuran

Setelah proses pencampuran selesai, akan didapatkan blok-blok hasil pencampuran yang akan menggantikan blok yang rusak. Citra hasil penyembunyian *error* diperlihatkan dalam Gambar 18.



Gambar 18. Citra hasil perbaikan 'foreman.jpg' (256 x 256 piksel)

Pada penilaian objektif digunakan PSNR untuk menilai kinerja metode MDI. Nilai PSNR yang diamati adalah perubahan nilai PSNR sebelum proses penyembunyian *error* dengan setelah proses penyembunyian *error*. Hasilnya disajikan dalam Tabel 2 dan Tabel 3.

TABEL 2. PERBANDINGAN NILAI PSNR UNTUK CITRA UKURAN 256 x 256 PIKSEL

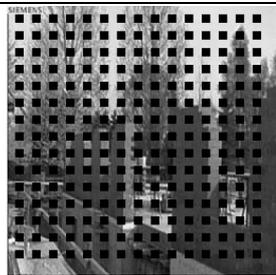
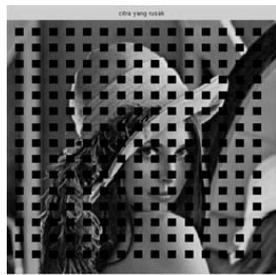
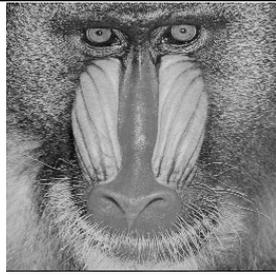
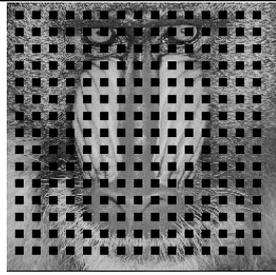
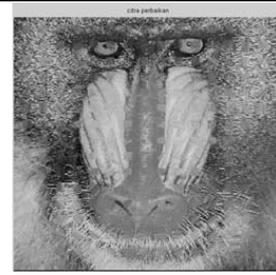
Nama citra	PSNR sebelum penyembunyian error	PSNR sesudah penyembunyian error	Kenaikan PSNR
Foreman.jpg	11,12dB	26,61dB	15,49dB
Taman.jpg	12,4dB	25,48dB	13,07dB
Lena.tif	13,94dB	27,57dB	13,63dB
Baboon.tif	12,03dB	24,25dB	12,22dB

TABEL 3. PERBANDINGAN NILAI PSNR UNTUK CITRA UKURAN 512 x 512 PIKSEL

Nama citra	PSNR sebelum penyembunyian error	PSNR setelah penyembunyian error	Kenaikan PSNR
Foreman.jpg	10,75dB	28,17dB	17,42dB
Taman.jpg	12,21dB	27,33dB	15,11dB
Lena.tif	13,71dB	27,35dB	13,64dB
Baboon.tif	11,75dB	24,79dB	13,04dB

Dari Tabel 2 dan Tabel 3 dapat dilihat bahwa kenaikan nilai PSNR setelah penyembunyian *error* menandakan blok-blok yang rusak dapat disembunyikan atau ditutupi dengan baik. Tabel 4 memperlihatkan hasil penyembunyian *error* untuk citra lain yang memiliki ukuran 256 x 256 piksel.

TABEL 4. CONTOH CITRA LAIN HASIL PENYEMBUNYIAN *ERROR* (256 x 256 PIKSEL)

Nama citra	Tanpa error	Citra yang rusak	Citra hasil perbaikan
Taman.jpg			
Lena.tif			
Baboon.tif			

XI. KESIMPULAN

Berdasarkan penilaian secara objektif, maka metode penyembunyian *error* dengan *Multi Directional Interpolation* secara keseluruhan terlihat tepi-tepi pada citra keluaran (hasil penyembunyian *error*) berkesinambungan (tidak diskontinu), sehingga tampilan objek pada citra terlihat jelas (blok yang cacat dapat disembunyikan).

UCAPAN TERIMA KASIH

Melalui tulisan ini Penulis juga ingin mengucapkan terima kasih kepada Sri Indriantanti, ST. yang telah membantu Penulis di dalam mengerjakan penelitian untuk tulisan ini.

DAFTAR PUSTAKA

- [1] R. Munir, *Pengolahan Citra Digital dengan Pendekatan Algoritmik*, Informatika, 2004.
- [2] W. Kwok, and H. Sun, "Multi-directional Interpolation for Spatial Error Concealment", *IEEE Transactions on Consumer Electronics*, vol. 39, no. 3, Agustus 1993, hal. 455– 460.
- [3] L. J. Galbiati, *Machine Vision and Digital Image Processing*, Englewood Cliff: Prentice-Hall, Inc., 1990.