

BAB III Metodologi Penelitian

Bab ini menjelaskan mengenai langkah-langkah yang dilakukan untuk membuat aplikasi konversi citra graylevel menjadi biner. Langkah-langkah yang diperlukan antara lain studi literatur, perancangan sistem, penentuan batasan algoritma yang digunakan, implementasi, pengujian dan analisis, pengambilan kesimpulan dan saran, serta penulisan laporan.

3.1 Studi Literatur

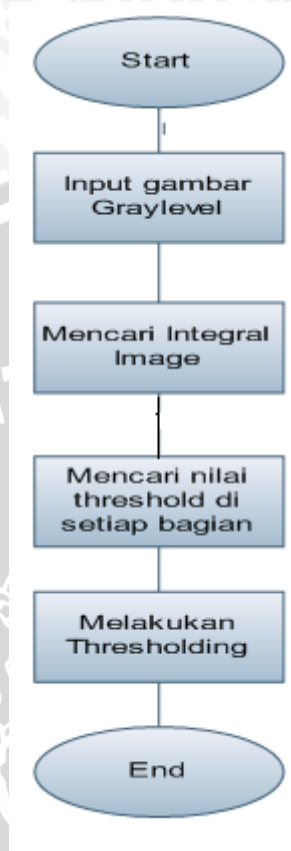
Berupa kajian pustaka yang mendukung pembuatan aplikasi yang berupa :

1. Kajian pustaka mengenai stuktur citra digital.
2. Kajian pustaka mengenai metode integral image
3. Kajian pustaka mengenai adaptive thresholding.
4. Kajian pustaka mengenai struktur dan bahasa pemrograman yang digunakan yaitu bahasa pascal, dalam hal ini Delphi.

3.2 Perancangan

Secara garis besar perancangan terdiri dari perancangan sistem dan cara kerja sistem,. Perancangan sistem yaitu merancang blok sistem diagram untuk menggambarkan sistem secara global dan selanjutnya dilakukan analisis sesuai kebutuhan. Analisis kebutuhan digunakan untuk menentukan kebutuhan-kebutuhan fungsi, algoritma yang digunakan, dan batasan perancangan dan implementasi. Berikut ini adalah alur utama dari system :





Gambar 3.1 Diagram Cara Kerja Sistem

Cara kerja sistem:

1. Program mempunyai masukan data berupa gambar Grayscale.
2. Mencari Integral Image dari gambar masukan
3. Membagi gambar menjadi beberapa bagian yang kemudian dicari nilai thresholdnya
4. Melakukan proses thresholding di setiap bagian gambar.

3.2.1 Input Gambar Grayscale

Untuk mendapatkan gambar yang akan di thresholding maka kita bisa memakai gambar berformat bitmap (*.bmp) graylevel (8 bit) yang ada pada hardisk yang dapat diakses melalui aplikasi.

Untuk menampilkan gambar pada aplikasi tidak berbeda dengan operasi open file pada windows, yaitu dengan masuk pada folder dimana gambar tersebut berada kemudian memilih file gambarnya,

3.2.2 Mencari Nilai Integral pada Gambar

Integral Image memiliki dimensi yang sama dengan dimensi gambar inputnya. Hanya saja pada Integral Image nilai tiap pixelnya merupakan penjumlahan seluruh pixel sebelah kiri dan sebelah atas. Jadi apabila intensitas pixel suatu gambar input dinyatakan dengan P dan nilai dari integral gambar tersebut adalah I , maka untuk mencari nilai $I(x,y)$ adalah sebagai berikut:

$$\begin{aligned}
 I(x,y) = & P(0,0)+P(1,0)+\dots+P(x,0)+ \\
 & P(0,1)+P(1,1)+\dots+P(x,1)+ \\
 & \dots \\
 & P(0,y)+P(1,y)+\dots+P(x,y)
 \end{aligned}$$

Untuk lebih jelasnya kita perhatikan contoh berikut:

		Gambar Input (P)				Integral Image(I)			
		0	1	2	3	0	1	2	3
Y	X								
	0	3	5	5	2	3	8	13	15
1	1	0	3	8	4	9	17	27	
2	2	6	7	9	6	17	32	51	
3	0	6	9	9	6	23	47	75	

Gambar 3.2 Pixel Gambar Input dan Integral Imagenya

Jadi untuk mencari nilai $I(1,1) = P(0,0) + P(1,0) + P(0,1) + P(1,1)$

$$= 3 + 5 + 1 + 0$$

$$= 9$$

3.2.3 Mencari Nilai Threshold

Konsep yang biasa digunakan untuk mencari nilai threshold adalah dengan mencari nilai rata-rata dari suatu area pada gambar yang kemudian nanti akan dibandingkan dengan nilai pixel itu sendiri.

Nilai Threshold sangat bergantung pada luasan area yang akan dicari rata-ratanya. Luasan ini biasanya berbentuk persegi $s \times s$ dimana nilai s yang disarankan adalah $1/8$ dari lebar gambar input (Pierre D. Wellner: *“Adaptive thresholding for the Digitaldesk”*). Nilai s yang terlalu kecil akan menghasilkan hasil thresholding yang buruk pada area gambar dengan kontras rendah, begitu juga nilai s yang terlalu besar, akan menghasilkan hasil thresholding yang buruk pada gambar berkontras yang tinggi.

Integral image mempermudah untuk mencari total nilai pixel di suatu area. Dengan menggunakan Integral Image kita bisa menghitung total pixel untuk area yang saling bertumpuk dengan jumlah operasi yang konstan. Berikut ini implementasi integral image untuk menghitung total pixel di suatu area dengan skala yang kecil.

		Gambar Input (P)				Integral Image(I)			
		0	1	2	3	0	1	2	3
Y	X								
	0		3	5	5	2	3	8	13
1		1	0	3	8	4	9	17	27
2		2	6	7	9	6	17	32	51
3		0	6	9	9	6	23	47	75

Gambar 3.3 Implementasi Integral Image

Misalkan kita memiliki gambar input dan telah didapatkan integral imagenya, dan kita akan menghitung total pixel di area yang berwarna abu-abu. Kita misalkan koordinat maksimum dari area tersebut adalah (x_2, y_2) dan koordinat minimumnya (x_1, y_1) maka dari gambar daerah yang berwarna abu-abu akan didapat :

$$x_1=2 \qquad y_1=2$$

$$x_2=3 \qquad y_2=3$$

Maka dengan persamaan:

$$\sum_{x=x_1}^{x_2} \sum_{y=y_1}^{y_2} f(x,y) = I(x_2,y_2) - I(x_2,y_1 - 1) - I(x_1 - 1,y_2) + I(x_1 - 1,y_1 - 1).$$

$$= I(3,3) - I(3,1) - I(1,3) + I(1,1)$$

$$= 75 - 27 - 23 + 9$$

$$= 34$$

Misalkan kita menghitung luasan pixel yang lain:

Gambar Input (P)					Integral Image(I)				
X \ Y	0	1	2	3	0	1	2	3	
0	3	5	5	2	3	8	13	15	
1	1	0	3	8	4	9	17	27	
2	2	6	7	9	6	17	32	51	
3	0	6	9	9	6	23	47	75	

Gambar 3.4 Implementasi Integral Image

$$x1=1$$

$$y1=1$$

$$x2=2$$

$$y2=2$$

Maka dengan persamaan:

$$\begin{aligned}
 \sum_{x=x_1}^{x_2} \sum_{y=y_1}^{y_2} f(x,y) &= I(x_2,y_2) - I(x_2,y_1 - 1) - I(x_1 - 1,y_2) + I(x_1 - 1,y_1 - 1). \\
 &= I(2,2) - I(2,0) - I(0,2) + I(0,0) \\
 &= 32 - 13 - 6 + 3 \\
 &= 16
 \end{aligned}$$

3.2.4 Melakukan Thresholding

Konsep thresholding adalah dengan mencari nilai rata-rata dari suatu area pada gambar yang kemudian nanti akan dibandingkan dengan nilai pixel itu sendiri. Tapi dalam aplikasi ini dilakukan sebaliknya, yaitu nilai pixel yang akan di threshold dikalikan jumlah pixel dalam luasan tersebut dibandingkan dengan total nilai pixel pada luasan yang sama.

Jadi bila suatu pixel dalam gambar input dilambangkan dengan $P(x,y)$, gambar output dilambangkan dengan $Out(x,y)$, dan total nilai pixel dalam suatu area $s \times s$ dilambangkan dengan sum , maka untuk melakukan thresholding dilakukan algoritma sebagai berikut:

If $P(x,y) \times (s \times s) \leq sum$ then

$Out(x,y) = 0$

Else $Out(x,y) = 255$

Nilai sum merupakan total nilai pixel yang didapat dengan menjumlahkan seluruh piksel di area $s \times s$ dengan menggunakan metode integral image. Hal ini memungkinkan untuk dilakukannya thresholding dengan area-area yang saling bertumpuk dan mudah dalam melakukan perubahan terhadap luas area.

3.3 Implementasi

Proses selanjutnya adalah implementasi yakni, proses transformasi hasil perancangan perangkat lunak yang telah dibuat ke dalam kode (*coding*) sesuai dengan sintaks dari bahasa pemrograman yang digunakan yaitu menerapkan bahasa pemrograman Pascal dengan menggunakan Borland Delphi 7.

3.4 Pengujian dan Analisis

Pengujian aplikasi yaitu untuk mengetahui kesesuaian analisa kebutuhan yang dibuat dengan implementasi aplikasi. Analisis sistem dilakukan dengan membandingkan sistem aplikasi yang dengan teori yang ada sehingga didapatkan suatu kesimpulan.

3.5 Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi dan pengujian sistem aplikasi telah selesai dilakukan dan didasarkan pada kesesuaian antara teori dan praktek. Kesimpulan diambil untuk menjawab rumusan masalah yang telah ditetapkan sebelumnya. Tahap yang paling akhir adalah pengambilan saran terhadap penelitian yang selanjutnya, sehingga bisa menyempurnakan kekurangan yang ada dan mengembangkan pada tingkat pokok kajian yang lebih lanjut.