

BAB IV PERANCANGAN

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan yang dilakukan meliputi perancangan sistem dan cara kerja system.

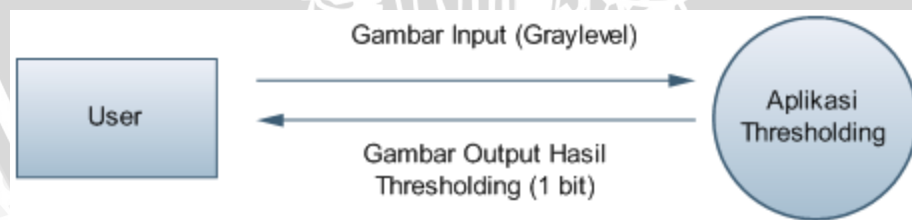
4.1 Perancangan Sistem

Perancangan sistem merupakan tahap awal dari perancangan perangkat lunak. Perancangan ini dilakukan untuk mengetahui aplikasi sistem yang akan dibuat secara umum. Perancangan sistem ini akan didahului dengan pendefinisian pelaku atau user yang akan menggunakan program ini dan juga alat bantu yang digunakan, yaitu:

1. User : Pelaku yang akan mengambil citra dan juga menjalankan program dengan tujuan untuk melakukan thresholding pada gambar graylevel.
2. Program : Perangkat lunak yang digunakan untuk merubah gambar graylevel (8 bit) menjadi gambar biner (1 bit) (citra masukan dan keluaran berformat bitmap (*.bmp)).

4.1.1 Diagram Konteks

Diagram konteks adalah sebuah diagram sederhana yang menggambarkan hubungan dengan entitas luar, masukan dan keluaran dari sistem. Diagram konteks aplikasi penyisipan gambar dengan enkripsi ditunjukkan pada gambar 4.1



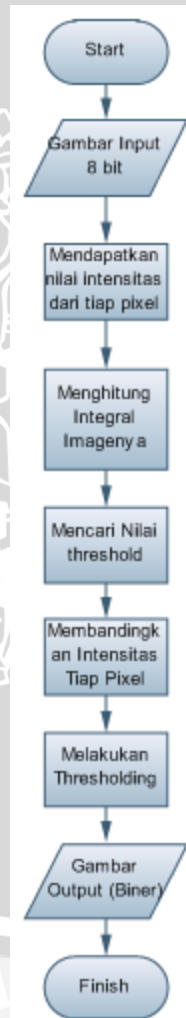
Gambar 4.1 Diagram Konteks

4.2 Cara Kerja Sistem

Cara kerja aplikasi penyisipan gambar adalah sebagai berikut:

1. Program mempunyai masukan data berupa citra graylevel.

2. Dari gambar input tersebut didapatkan intensitas tiap pixelnya untuk kemudian dihitung integral imagenya.
3. Hasil penghitungan integral Image disimpan dalam array dua dimensi (dimensi array integral image sama dengan dimensi gambar input).
4. Dilakukan penghitungan nilai threshold untuk tiap pixel dengan menggunakan integral image.
5. Membandingkan intensitas pixel dengan nilai thresholdnya
6. Melakukan thresholding. Bila intensitas pixel gambar input lebih dari nilai thresholdnya maka warna pixel tersebut pada gambar keluaran adalah putih, dan bila sebaliknya maka warna pixel gambar keluarannya adalah hitam.



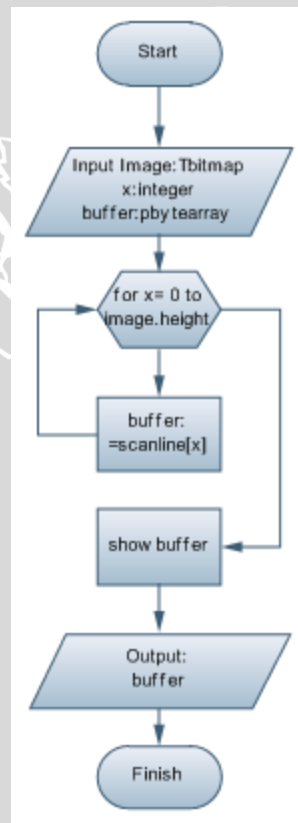
Gambar 4.3 Diagram Alir Sistem Secara Umum

4.2.1 Input Gambar

Dalam Aplikasi ini, untuk menginput gambar, user disediakan sebuah tombol pada tampilan utama aplikasi. Bila tombol tersebut ditekan maka kita bisa memilih gambar mana yang ingin kita pakai. Setelah memilih file gambar, maka ditampilan utama akan muncul gambar input yang akan di thresholding.

4.2.2 Mendapatkan Nilai intensitas Pixel

Ada beberapa metode untuk mendapatkan nilai intensitas pixel dari suatu gambar pada Delphi. Dalam aplikasi ini kita akan menggunakan property scanline. Dengan menggunakan scanline kita bisa mendapatkan nilai pixel dari suatu gambar secara perbaris, karena itu relatif lebih cepat dibandingkan dengan mengakses gambar secara perpixel. Adapun algoritma dalam menampilkan nilai pixel dari gambar adalah sebagai berikut:



Gambar 4.4 Diagram Alir menampilkan intensitas pixel

Setelah mendapatkan nilai intensitas tiap pixel dari gambar, maka akan mempermudah untuk melakukan manipulasi gambar lebih jauh. Selain itu juga memungkinkan dilakukannya penghitungan integral image dari gambar input.

4.2.3 Menghitung Integral Image

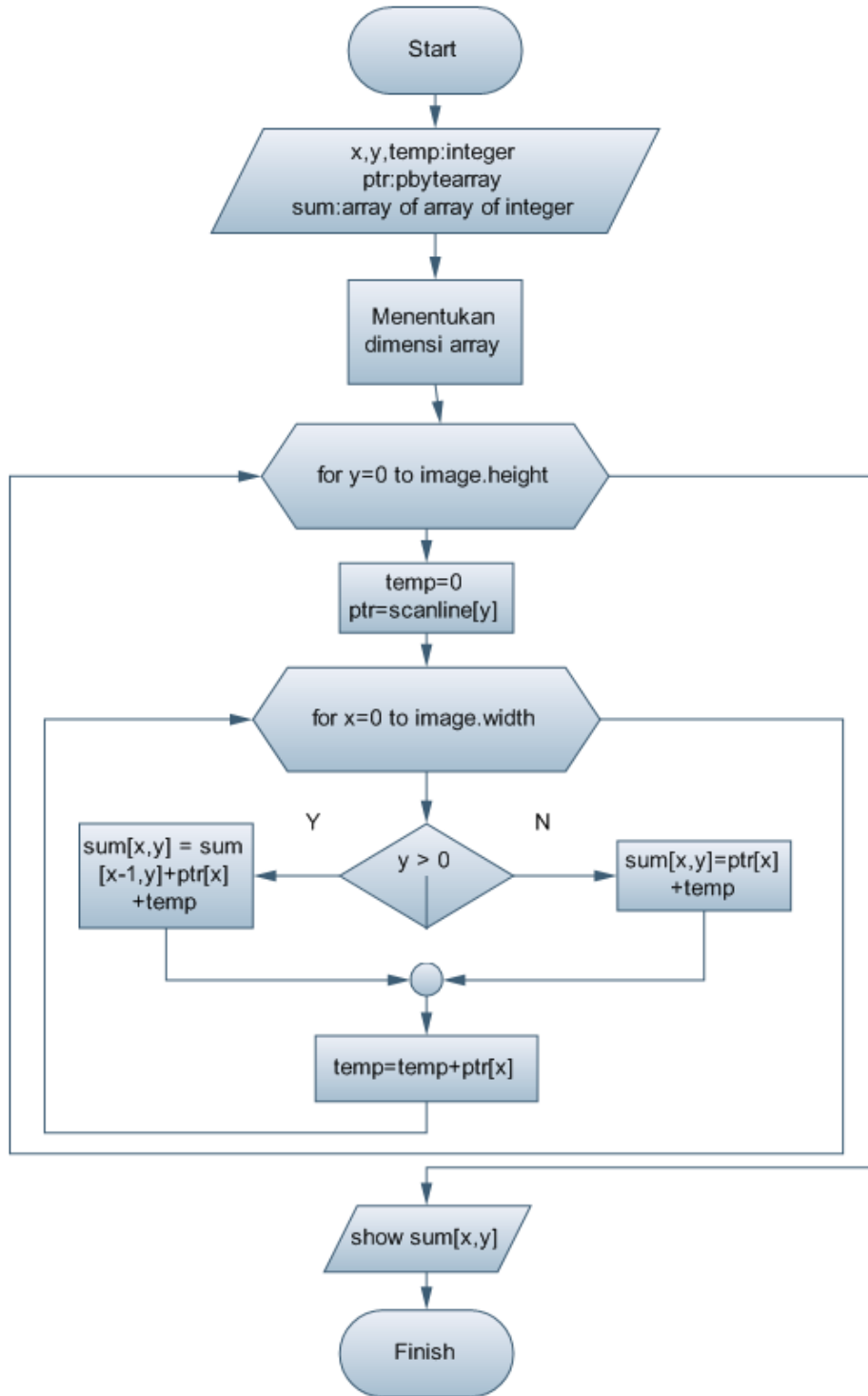
Karena scanline membaca mulai dari kolom pixel paling kiri kemudian ke kanan, maka untuk mencari integral image pada baris pertama gambar input hanya perlu dengan menjumlahkan pixel yang telah di lalui dan piksel saat ini. Sedangkan untuk baris selanjutnya, sama seperti pada baris pertama hanya perlu ditambahkan dengan hasil integral image untuk pixel di atasnya. Jadi misalkan intensitas gambar dinotasikan sebagai P dan integral image sebagai I maka:

$$I(x,y) = \sum_{x=0}^{x-1} f(x,y) + I(x,y-1) + P(x,y)$$

		Gambar Input (P)				Integral Image(I)			
		0	1	2	3	0	1	2	3
X \ Y	X								
	Y								
	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 4.5 Pixel Gambar Input dan Integral Imagenya

Nilai- nilai integral image merepresentasikan hasil penjumlahan seluruh pixel gambar input di sebelah kiri dan atasnya. Karena aplikasi melintasi pixel gambar mulai dari kiri ke kanan setelah itu kebawah. Untuk lebih jelasnya bisa kita perhatikan pada gambar 4.5, saat kita akan mencari integral image pada x,y=(2,2) maka untuk nilai I(2,1) sudah merepresentasikan hasil seluruh penjumlahan dari P(0,0) hingga P(2,1), tetapi kita masih memerlukan hasil penjumlahan antara I(0,2) hingga I(2,2). I(2,2) merupakan nilai piksel saat ini dan kita membutuhkan sebuah variable untuk menampung nilai yang merupakan hasil penjumlahan pixel dari baris ke-2 mulai x=0 hingga x=x-1. Untuk lebih jelasnya bisa kita lihat diagram alir berikut:



Gambar 4.6 Diagram Alir menghitung integral image

4.2.4 Mencari Nilai Threshold

Dalam aplikasi ini, untuk mencari nilai threshold tidak diperlukan untuk mencari rata-rata

nilai pixel di suatu area gambar, melainkan cukup dengan menjumlah semua nilai piksel di suatu area (tidak perlu dibagi jumlah pixelnya). Hal ini dikarenakan pada saat membandingkan dengan nilai intensitas suatu pixel, pixel tersebut akan dikalikan dengan jumlah pixel di area yang diinginkan.

Metode integral image sangat efektif untuk digunakan dalam mencari nilai total intensitas dalam suatu luasan gambar. Karena bisa dihasilkan dengan proses penghitungan yang konstan (tidak dipengaruhi jumlah piksel dalam suatu luasan) dan luas area yang dinamis.

Untuk mencari nilai threshold harus ditentukan terlebih dahulu daerah luasan yang diinginkan, tentu saja daerah luasan ini tidak terlalu besar dan juga tidak terlalu kecil. Daerah tersebut berbentuk persegi dengan nilai pixel yang akan dibandingkan dengan nilai threshold sebagai pusatnya. Untuk selanjutnya kita notasikan panjang (yang juga sama dengan lebar, karena berbentuk persegi) luasan ini dengan s , sehingga luas daerah tersebut adalah $s \times s$. Berikut ini adalah pseudocode untuk menjelaskan pencarian total pixel di suatu luasan.

```

for x = 0 to w do
  for y = 0 to h do
    x1    =i-s/2 {border checking is not shown}
    x2    =i+s/2
    y1    =j-s/2
    y2    =j+s/2
    count = (x2-x1)x(y2-y1)
    sum   =intImg[x2,y2]-intImg[x2,y1-1]-intImg[x1-1,y2]+intImg[x1-1,y1-1]
  
```

dimana:

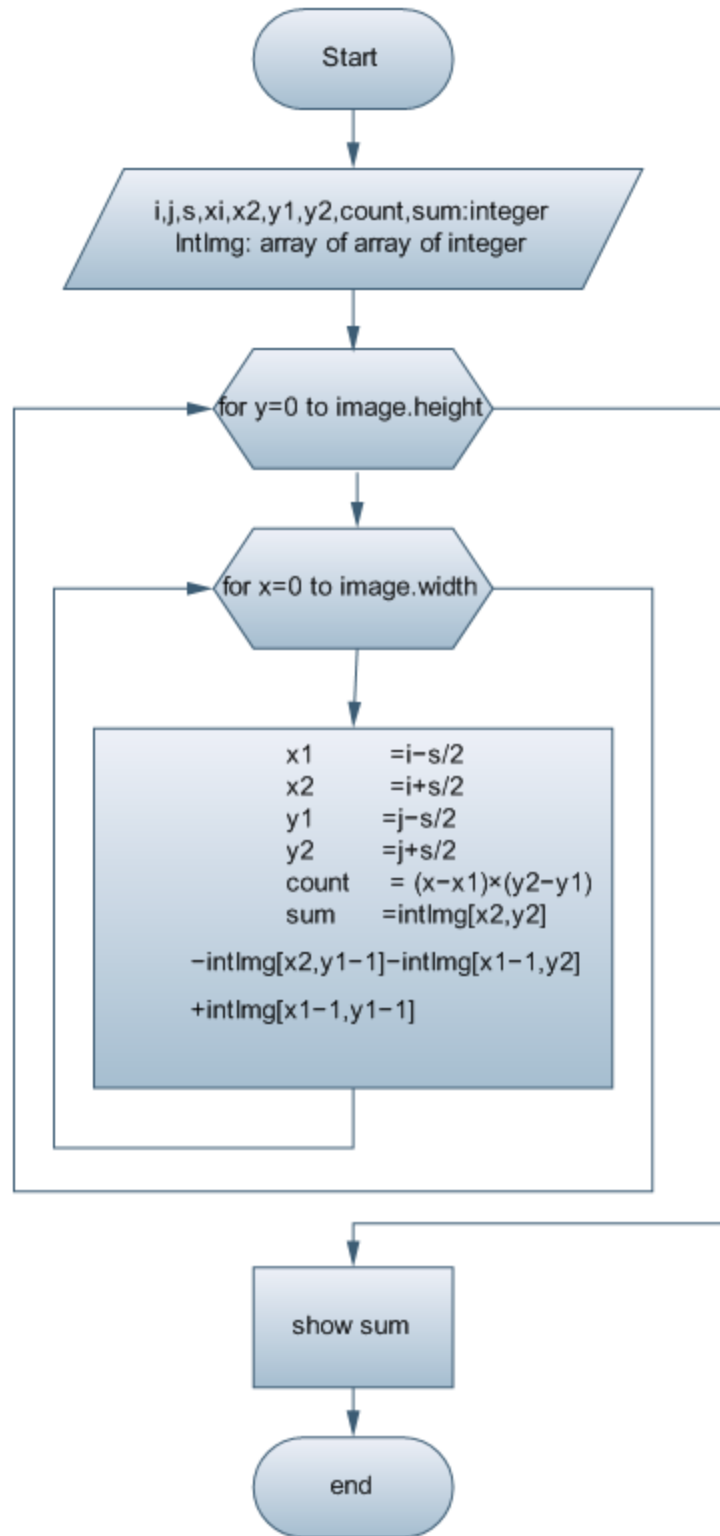
w : lebar image.

h : tinggi image.

Count : jumlah pixel di area luasan.

Sum : total nilai pixel di area luasan.

IntImg : nilai integral image.



Gambar 4.7 Diagram Alir menghitung integral image

4.2.5 Melakukan Thresholding

Thresholding merupakan suatu proses konversi citra dari citra graylevel menjadi citra biner. Proses thresholding ini dilakukan dengan membandingkan nilai intensitas suatu pixel dengan nilai thresholdnya. Apabila nilainya lebih rendah dari nilai thresholdnya maka nilai intensitas gambar keluaran pada pixel tersebut adalah 0 (hitam), begitu juga sebaliknya, apabila nilai intensitas pixel tersebut lebih besar dari nilai thresholdnya maka output gambar pada pixel tersebut adalah 255 (putih).

Yang sedikit berbeda dalam aplikasi ini adalah perbandingan nilai intensitas pixel dengan nilai thresholdnya. Apabila pada umumnya nilai threshold adalah rata-rata nilai suatu pixel dalam suatu area, maka pada aplikasi ini nilai threshold merupakan nilai total piksel dalam suatu area. Oleh karena itu nilai intensitas piksel harus dikalikan dengan jumlah piksel di area yang sama untuk mendapatkan hasil thresholding yang baik.

```

for x = 0 to image.width do
  for y = 0 to image.height do
    if (in[x,y]×count) < (sum×(100-t)/100) then
      out[x,y] = 0
    else
      out[x,y] = 255
    end if
  end for
end for

```

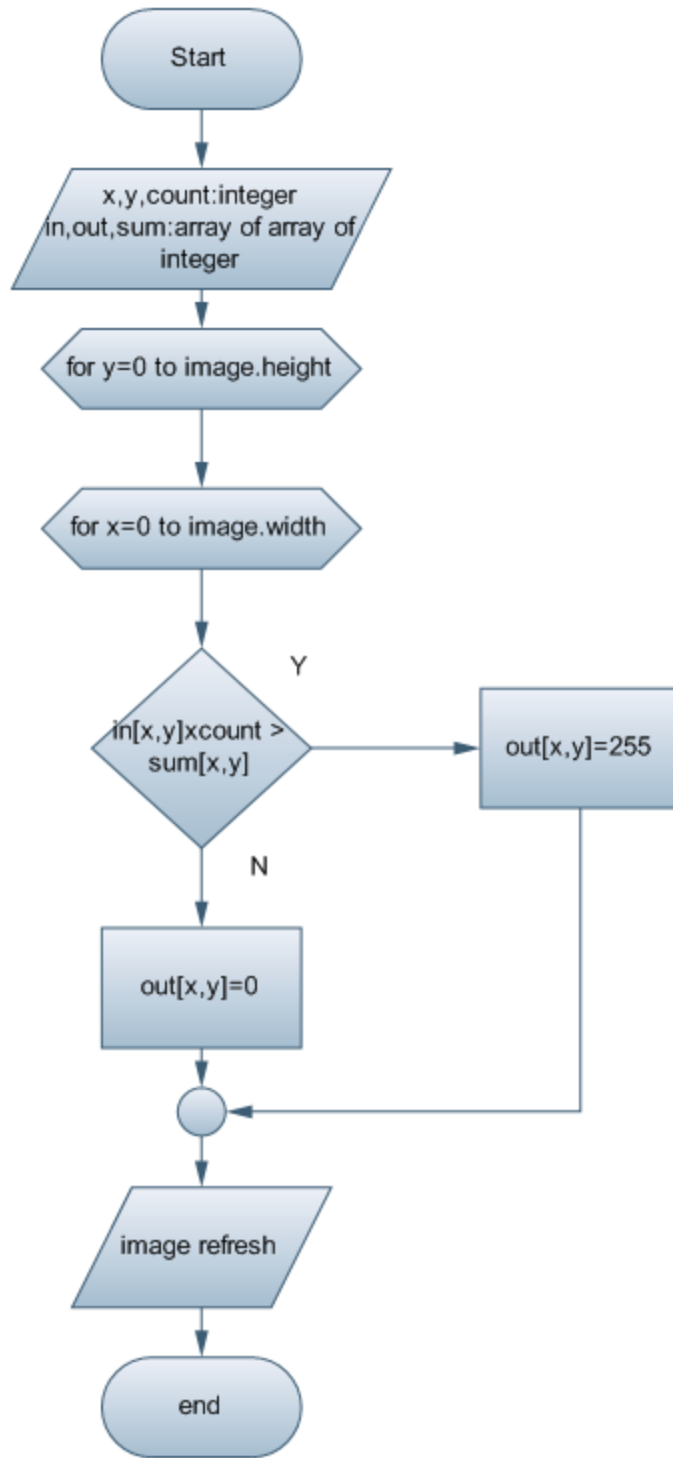
keterangan:

in : intensitas pixel gambar input.

count : jumlah piksel di suatu area (s x s).

sum : jumlah total pixel di suatu area ($\sum(in)$).

Out : intensitas gambar keluaran.



Gambar 4.8 Diagram Alir Thresholding

