

## Pengenalan Bentuk Bangun Datar Menggunakan Kode Rantai

Kholifatun Nisa, Tamam Asrori, Misdiyanto

Fakultas Teknik, Program Studi Teknik Elektro  
Universitas Panca Marga, Jln. Yos Sudarso 107 Pabean Dringu Probolinggo 67271.

Email: [kholifatun\\_nisa@rocketmail.com](mailto:kholifatun_nisa@rocketmail.com)

Terima Naskah : 28 April 2014

Terima Revisi : 05 Mei 2014

### ABSTRAK

*Computer vision* adalah kombinasi antara pengolahan citra dan pengenalan pola. Pengolahan citra digital digunakan untuk memperbaiki dan mengolah informasi suatu gambar untuk keperluan pengenalan objek. Dalam skripsi ini, sistem pengenalan bentuk bangun datar menggunakan kode rantai dibuat dengan menggunakan *software* Matlab. Bentuk bangun datar yang digunakan antara lain segitiga sama sisi, segi empat, jajaran genjang dan trapesium. Perancangan dilakukan dengan membuat sistem pengenalan yang memiliki beberapa tahap diantaranya adalah tahap membaca citra hasil citraan kamera digital, memotong citra, mengubah ukuran citra menjadi seragam atau sama, mengubah jenis warna citra dari RGB menjadi abu-abu, mencari batas tepi objek pada citra menggunakan teknik konvolusi dengan operator sobel, mengubah citra hasil konvolusi menjadi citra biner, melakukan penipisan tepi objek sehingga menjadi setebal satu piksel, mendeteksi kode rantai objek, melakukan pengenalan pola dengan mencocokkan kode rantai acuan yang telah disimpan sebelumnya dengan kode rantai pada citra baru, dan terakhir yaitu mendeskripsikan bentuk objek. Dari hasil pengujian terhadap sistem, diketahui bahwa sistem dapat membaca semua citra uji. Kemudian proses pemotongan citra, pengubahan ukuran citra, pengubahan warna, pendeteksian tepi, binerisasi, penipisan tepi objek, dan deteksi kode rantai, semuanya dapat berjalan sesuai yang diharapkan. Persentase keberhasilan pada implementasi kode rantai untuk pengenalan bangun datar adalah 96,875%.

*Kata kunci: Pengolahan Citra, Pengenalan Pola, Bangun Datar, Kode Rantai.*

### ABSTRACT

*Computer vision is combination of image processing and pattern recognition. Digital Image Processing used for fix and process image information for identifying object. In this paper, two dimensions shape identification system use chain code made from Matlab. two dimensions shape used is equilateral triangle, rectangle, parallelogram, and trapezium. Design did by makes recognition system that has many stage such as read image from digital camera, cropping image, resize image for the size same, change color of image to grayscale, find the image edge use konvolution with sobel, change konvolution image to image binary, thinning the edge become one pixel, find the chain code, do the pattern recognition by check the reference chain code with new image chain code, and the last is describe a shape of object. From the system testing, system can read all image tests. then the cropping image, resize image, grayscaling, edge detection, change image to image binary, thinning, and chain code detection, all of them gives expected results. Success percentage on chain code implementation for shape identification is 96,875%.*

*keywords: Image Processing, Pattern Recognition, Two Dimension Shape, Chain Code.*

## PENDAHULUAN

Seiring dengan perkembangan zaman, teknologi juga berkembang dengan pesat. Begitu pula dengan perkembangan komputer. Hampir setiap individu memerlukan komputer sebagai alat bantu untuk menyelesaikan masalah. Hampir semua sistem analog diganti dengan sistem komputerisasi. Ini dilakukan karena sistem komputerisasi lebih mudah dalam pengontrolannya. Dalam hal ini misalnya pengontrolan dalam mengidentifikasi suatu citra. Komputer diusahakan untuk dapat bekerja mendekati proses kerja otak manusia.

Kemampuan komputer yang dapat meniru kerja otak manusia ini dalam bidang ilmu komputer disebut kecerdasan buatan. Kecerdasan buatan membuat komputer lebih berguna dan mampu melakukan tugas-tugas yang cerdas secara efektif dan efisien. Kecerdasan buatan banyak berkaitan dengan berbagai bidang, salah satunya adalah *computer vision*. *Computer vision* mempunyai kemampuan untuk menganalisis objek secara visual, setelah data objek dikonversi dalam bentuk citra digital.

*Computer vision* adalah kombinasi antara pengolahan citra dan pengenalan pola. Pengolahan citra merupakan proses awal dari *computer vision*, dan pengenalan pola merupakan proses menginterpretasikan citra. Sedangkan teknik-teknik yang digunakan untuk mengolah citra disebut pengolahan citra digital. Pengolahan citra digital digunakan untuk memperbaiki kualitas suatu gambar sehingga dapat dengan mudah diinterpretasikan oleh mata manusia dan untuk mengolah informasi yang terdapat pada suatu gambar untuk keperluan pengenalan objek secara otomatis. Untuk mengenali objek secara otomatis, maka dibutuhkan suatu teknik, teknik ini akan melakukan pengenalan sehingga menghasilkan informasi yang dibutuhkan. Teknik ini disebut kode rantai. Kode rantai menghasilkan suatu kode yang menunjukkan arah pergerakan dari perbatasan luar suatu objek yang saling menyambung membentuk rantai. Objek yang digunakan adalah objek hasil citraan yang berupa objek dua dimensi.

Kemampuan inilah yang akan peneliti tiru dengan menggunakan komputer. Komputer akan menerima masukan berupa citra, kemudian citra tersebut diproses dan akan memberikan keluaran berupa deskripsi objek pada citra. Objek yang peneliti gunakan adalah empat bentuk bangun

datar yaitu segitiga sama sisi, segi empat, jajaran genjang, dan trapesium.

Berdasarkan uraian yang telah dijelaskan, maka peneliti menyusun penelitian dengan judul "Pengenalan Bentuk Bangun Datar menggunakan Kode Rantai". Penelitian ini membahas cara mengimplementasikan kode rantai untuk pengenalan bentuk bangun datar.

## METODE PENELITIAN

Pada penelitian ini, objek yang akan diidentifikasi yaitu bentuk bangun datar yang terdiri dari segitiga sama sisi, segi empat, jajaran genjang, dan trapesium.

### Perangkat Penelitian

Perangkat yang digunakan pada penelitian ini ada dua macam yaitu *Hardware* dan *Software*. Adapun *Hardware* yang digunakan adalah sebagai berikut.

1. Kamera Canon Power Shot A2500.
2. Laptop Samsung AMD E1-1200 APU with Radeon(tm) HD Graphics 1.40 GHz .

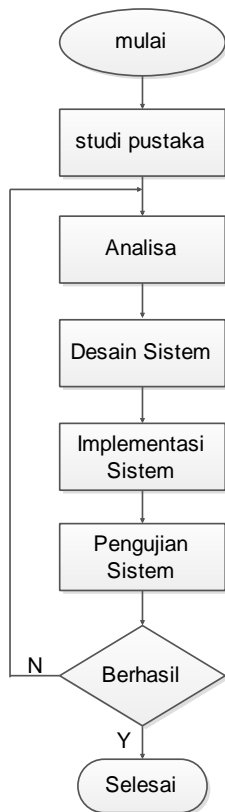
Sedangkan *software* yang digunakan adalah sebagai berikut.

1. Windows 7.
2. Matlab R2007b.

### Metode Pengumpulan data

Metode pengumpulan data yang peneliti gunakan adalah sebagai berikut.

1. Metode Studi Pustaka  
Metode studi pustaka dilakukan dengan mengumpulkan data dari buku-buku yang sesuai dengan tema penelitian.
2. Metode Studi Dokumen  
Metode studi dokumen dilakukan dengan mengumpulkan data dari literatur-literatur seperti jurnal, hasil penelitian, skripsi, dokumentasi dari internet dan sebagainya sebagai bahan penunjang dalam penelitian.



Gambar 1. Flowchart Rancangan Penelitian (Sumber: data diolah)

**Desain Sistem**

Desain diperlukan agar tahap-tahap yang dilakukan sesuai dengan hasil yang diharapkan. Berikut peneliti jelaskan tahap-tahap dari desain sistem tersebut.

1. Mengambil citra bangun datar berupa foto berekstensi jpg hasil citraan menggunakan kamera digital.
2. Memotong citra sehingga memiliki lebar dan tinggi yang sama.
3. Mengubah ukuran citra menjadi ukuran yang lebih kecil.
4. Mengubah citra asli/RGB menjadi citra abu-abu.
5. Mencari batas tepi objek dengan cara konvolusi dengan menggunakan operator sobel.
6. Mengubah citra hasil konvolusi menjadi citra biner.
7. Melakukan penipisan tepi objek menggunakan proses *thinning*.
8. Mencari kode rantai dari objek menggunakan kode rantai 8 lintasan dengan penandaan searah jarum jam.

9. Melakukan pengenalan pola bangun datar dengan mencocokkan nilai kode rantai acuan dengan nilai kode rantai citra masukan.
10. Deskripsi objek. Jika nilai kode rantai pada citra masukan sama dengan nilai kode rantai acuan maka akan menghasilkan keluaran berupa deskripsi bentuk bangun datar.



Gambar 2. Flowchart Sistem (Sumber: data diolah)

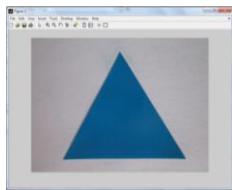
## HASIL DAN PEMBAHASAN

### Pengujian Sistem

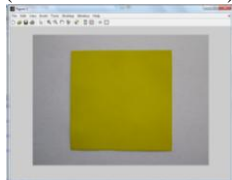
Proses pengujian ini memiliki beberapa tahap yang dilakukan, tahap-tahap tersebut diantaranya adalah sebagai berikut.

#### 1. Ambil Citra

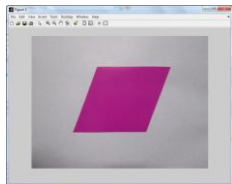
Ambil citra adalah tahap awal yang dilakukan dalam penelitian ini. Citra masukan berupa citra hasil citraan menggunakan kamera digital dengan berkas penyimpanan berekstensi jpg.



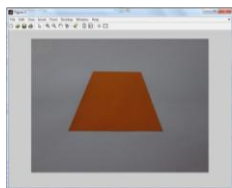
Gambar 3. Citra Segitiga  
(Sumber: data diolah)



Gambar 4. Citra Segi empat  
(Sumber: data diolah)



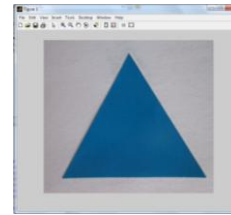
Gambar 5. Citra Jajaran Genjang  
(Sumber: data diolah)



Gambar 6. Citra Trapesium  
(Sumber: data diolah)

#### 2. Memotong Citra

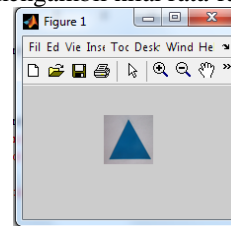
Langkah selanjutnya yang dilakukan setelah mengambil citra yaitu memotong citra (*Cropping*). Pemotongan dilakukan untuk mengubah ukuran piksel citra sehingga memiliki ukuran lebar dan tinggi yang sama yaitu 520 piksel x 520 piksel.



Gambar 7. Citra hasil *Cropping*  
(Sumber: data diolah)

#### 3. Mengubah Ukuran Citra

Seperti yang telah dijelaskan sebelumnya bahwa ukuran citra mempengaruhi hasil keluaran, karena sifat kode rantai yang sangat peka terhadap tepian objek. Oleh karena itu, sebelum masuk ke tahap berikutnya dilakukan tahap mengubah ukuran citra (*Resize*). Tahap ini mengubah ukuran piksel pada citra menjadi satu ukuran yang seragam atau sama yaitu 50 piksel x 50 piksel dengan cara mengambil nilai rata-rata piksel..



Gambar 8. Citra hasil *Resize*  
(Sumber: data diolah)

#### 4. Mengubah Citra Warna/RGB menjadi Citra Abu-abu

Tahap selanjutnya dalam penelitian ini adalah *grayscale*. Pada tahap ini, citra masukan yang berupa citra RGB diubah menjadi citra abu-abu. Proses tersebut dilakukan pada setiap piksel citra, dengan cara ini setiap piksel memiliki satu jenis warna dengan intensitas yang berbeda-beda.

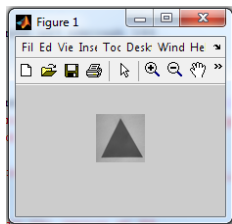
Seperti yang telah dijelaskan sebelumnya, bahwa citra RGB memiliki tiga layer warna yang berbeda yang memiliki nilai piksel pada masing-masing layer dan didefinisikan pada matriknya masing-masing. Untuk mengubahnya menjadi citra abu-abu maka ketiga matrik citra RGB dikalikan dengan nilai *gray strength* kemudian dijumlahkan, sehingga menjadi satu matrik tunggal yang menyimpan nilai piksel warna yang berkisar dari warna hitam sampai putih atau secara visual terlihat berwarna abu-abu. Nilai *gray strength* memiliki pengaruh terhadap hasil pengujian oleh karena itu harus ditetapkan nilai yang tepat untuk

menghasilkan keluaran yang sesuai. Nilai *gray strength* berkisar antara 0,1 sampai 0,99. Rumus mengubah citra RGB menjadi citra abu-abu adalah sebagai berikut.

$$s = (n_1 * r) + (n_2 * g) + (n_3 * b)$$

dimana:

- s adalah nilai citra abu-abu,
- n adalah nilai *gray strength*,
- r adalah nilai citra merah (red),
- g adalah nilai citra hijau (green), dan
- b adalah nilai citra biru (blue).



Gambar 9. Citra Abu-abu (Sumber: data diolah)

Dalam penelitian ini, peneliti menggunakan nilai *gray strength* berturut-turut adalah 0,3 untuk  $n_1$ , 0,3 untuk  $n_2$ , dan 0,4 untuk  $n_3$ . Dengan menggunakan nilai tersebut diperoleh hasil pengujian yang baik dibandingkan dengan menggunakan kombinasi nilai yang lain. Nilai *gray strength* ini memiliki pengaruh terhadap hasil uji bentuk tepi objek pada citra (citra hasil *thinning*). Hasil pengujian dapat dilihat pada tabel dibawah dengan nilai *threshold* yang sama yaitu 0,9.

Tabel 1. Hasil Pengujian Nilai Gray Strength

No	Nilai Gray Strength			Ket.
	$0.4*r+0.3*g+0.3*b$	$0.3*r+0.4*g+0.3b$	$0.3*r+0.3*g+0.4*b$	
1				Bangun segitiga berwarna merah
2				Bangun segi empat berwarna kuning
3				Bangun Jajaran Genjang berwarna hijau
4				Bangun trapesium berwarna biru

(Sumber: data diolah)

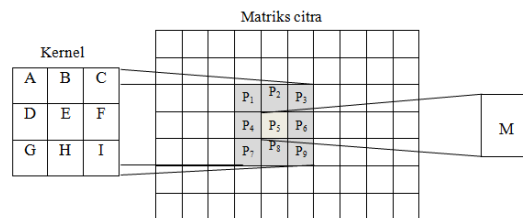
Dari tabel di atas dapat dilihat bahwa hasil uji citra dengan menggunakan kombinasi nilai *gray strength*  $0,3*r+0,3*g+0,4*b$  lebih bagus dari kombinasi nilai  $0,4*r+0,3*g+0,3*b$  maupun  $0,3*r+0,4*g+0,3*b$ .

**5. Tahap Pencarian Batas Tepi Objek**

Tahap pencarian batas tepi dilakukan melalui proses konvolusi dengan menggunakan operator sobel. Caranya dengan menggeser kernel piksel per piksel, kemudian hasilnya disimpan dalam matriks baru. Untuk matriks yang tidak dikonvolusi yaitu matriks pinggir maka nilainya tetap sama dengan nilai piksel semula. Operator sobel menggunakan pencarian tepi secara horizontal dan vertikal. Dalam penelitian ini, batas horizontal ( $G_y$ ) dan vertikal ( $G_x$ ) ditunjukkan seperti gambar 4.8 berikut.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Gambar 10. Jendela  $G_x$  dan  $G_y$  (Sumber: data diolah)



Gambar 11. Ilustrasi Konvolusi (Sumber: data diolah)

Untuk mencari tepi pada citra digunakan rumus konvolusi sebagai berikut.

$$M = \sqrt{S_x^2 + S_y^2}$$

$$S_x = AP_1 + DP_2 + GP_3 + BP_4 + EP_5 + HP_6 + CP_7 + FP_8 + IP_9$$

$$S_y = AP_1 + BP_2 + CP_3 + DP_4 + EP_5 + FP_6 + GP_7 + HP_8 + IP_9$$

Dalam hal ini, M adalah hasil konvolusi,  $P_1-P_9$  adalah piksel citra yang dikenai operasi,  $S_x$  adalah jumlah dari perkalian antara  $G_x$  dan piksel pada citra, dan  $S_y$  adalah jumlah dari perkalian antara  $G_y$  dan piksel citra. Operasi ini diilustrasikan pada Gambar berikut.

3	4	2	5	1
2	1	6	4	2
3	5	7	1	3
4	2	5	7	1
2	5	1	3	2

	18			

Gambar 12. Contoh Operasi Konvolusi (1)  
(Sumber: data diolah)

3	4	2	5	1
2	1	6	4	2
3	5	7	1	3
4	2	5	7	1
2	5	1	3	2

	18	10		

Gambar 13. Contoh Operasi Konvolusi (2)  
(Sumber: data diolah)

3	4	2	5	1
2	1	6	4	2
3	5	7	1	3
4	2	5	7	1
2	5	1	3	2

	18	10	14	

Gambar 14. Contoh Operasi Konvolusi (3)  
(Sumber: data diolah)

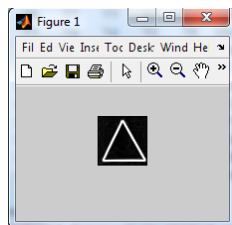
3	4	2	5	1
2	1	6	4	2
3	5	7	1	3
4	2	5	7	1
2	5	1	3	2

	18	10	14	
	16			

Gambar 15. Contoh Operasi Konvolusi (4)  
(Sumber: data diolah)

3	4	2	5	1
2	18	10	14	2
3	16	2	20	3
4	12	14	14	1
2	5	1	3	2

Gambar 16. Contoh Hasil Konvolusi  
(Sumber: data diolah)



Gambar 17. Citra hasil Konvolusi  
(Sumber: data diolah)

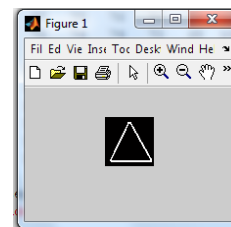
### 6. Mengubah Citra Hasil Konvolusi menjadi Citra Biner

Pada tahap ini setiap piksel yang ada diubah derajat keabuannya menjadi 0 untuk derajat keabuan terendah dan 1 untuk tingkat derajat keabuan tertinggi. Proses ini disebut juga binerisasi, karena citra yang dihasilkan adalah citra biner. Tahap *thresholding* ini dilakukan untuk memudahkan pada tahap penelusuran kode rantai.

3	4	2	5	1
2	18	10	14	2
3	16	2	20	3
4	12	14	14	1
2	5	1	3	2

0	0	0	0	0
0	1	1	1	0
0	1	0	1	0
0	1	1	1	0
0	0	0	0	0

Gambar 18. Contoh Proses *Thresholding*  
(Sumber: data diolah)















Gambar 19. Citra hasil *Thresholding*  
(Sumber: Data diolah)

Cara mengubah citra hasil konvolusi ke citra biner pada penelitian ini yaitu dengan memberi nilai *threshold* 0,9 atau 229,5/255. Dengan nilai tersebut, nilai citra konvolusi yang memiliki nilai di bawah 229,5 akan diubah menjadi nilai 0 dan nilai citra konvolusi di atas 229,5 akan diubah menjadi nilai 1. Alasan peneliti menggunakan nilai 0,9 karena nilai tersebut menghasilkan citra uji lebih baik dari citra uji dengan nilai *threshold* lain.

Selain nilai *gray strength*, nilai *threshold* juga memiliki pengaruh terhadap hasil uji bentuk tepi objek pada citra (citra hasil *thinning*). Hasil pengujian dapat dilihat pada tabel 4.2, dengan variasi nilai *threshold* yaitu 0,7, 0,8, dan 0,9.

Tabel 2. Hasil Pengujian Nilai *Threshold*

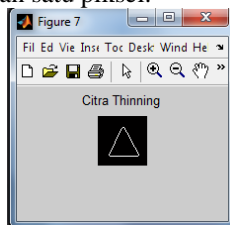
No	Nilai <i>Threshold</i>			Ket.
	0,7	0,8	0,9	
1				Bangun segitiga berwarna merah
2				Bangun segi empat berwarna kuning
3				Bangun Jajaran Genjang berwarna hijau
4				Bangun trapesium berwarna biru

(Sumber: data diolah)

Dari tabel diatas dapat diketahui bahwa hasil uji citra menggunakan nilai *threshold* 0,9 lebih bagus dibanding dengan hasil uji menggunakan nilai *threshold* 0,8 maupun 0,7.

**7. Penipisan Tepi Objek**

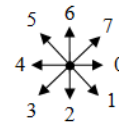
Tahap penipisan tepi objek atau disebut juga dengan *thinning* digunakan untuk mengurangi suatu daerah yang tebal atau bergumal pada suatu citra menjadi sebuah citra dengan piksel-piksel tunggal. Proses ini dapat dilakukan dengan cara mengurangi tiap-tiap piksel yang ada hingga mencapai ketebalan satu piksel.



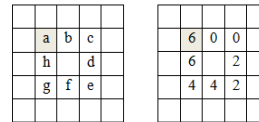
Gambar 20. Citra hasil *Thinning*  
(Sumber: data diolah)

**8. Penelusuran Kode Rantai**

Tahap penelusuran kode rantai pada penelitian ini menggunakan delapan arah kode rantai dengan arah penandaan searah dengan arah jarum jam. Gambar dibawah menunjukkan bentuk kode rantai yang peneliti gunakan. Untuk penentuan titik awal penelusuran kode rantai ini, dilakukan pelacakan piksel dari kiri ke kanan serta dari atas ke bawah dengan mencari piksel berwarna putih atau bernilai satu yang pertama kali ditemukan.



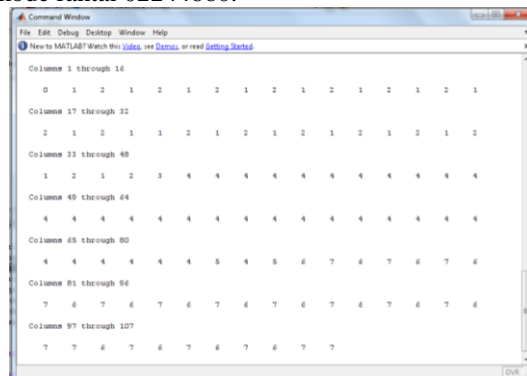
Gambar 21. Penomoran Arah Kode Rantai  
(Sumber: data diolah)



Gambar 22. Contoh Penelusuran Kode Rantai  
(Sumber: data diolah)

Seperti yang dicontohkan pada gambar d, dimisalkan piksel awal yang ditemukan adalah a. piksel awal ini masih belum memiliki arah sehingga kode rantai pertama yang ditulis yaitu kode rantai pada piksel berikutnya. Karena pencarian kode rantai searah jarum jam sehingga kode rantai pertama adalah b yaitu 0, c dengan kode rantai 0, d dengan kode rantai 2 dan seterusnya hingga kembali lagi ke piksel awal.

Dari contoh tersebut maka akan dihasilkan kode rantai 02244660.



Gambar 23. Contoh Hasil Pencarian Kode Rantai Citra Segitiga  
(Sumber: data diolah)

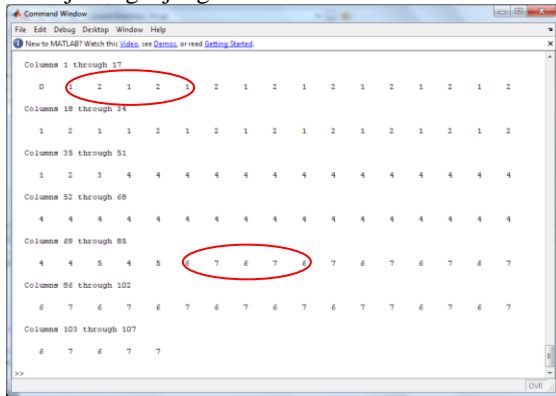
**9. Pengenalan Pola**

Tahap pengenalan merupakan tahap mencocokkan kode acuan suatu citra dengan kode rantai yang menjadi acuan. Metode pencocokan kode rantai adalah salah satu metode terapan dari teknik konvolusi. Caranya adalah dengan mencocokkan nilai kode rantai yang telah disimpan sebelumnya berdasarkan bentuk objek dengan kode rantai pada citra masukan.

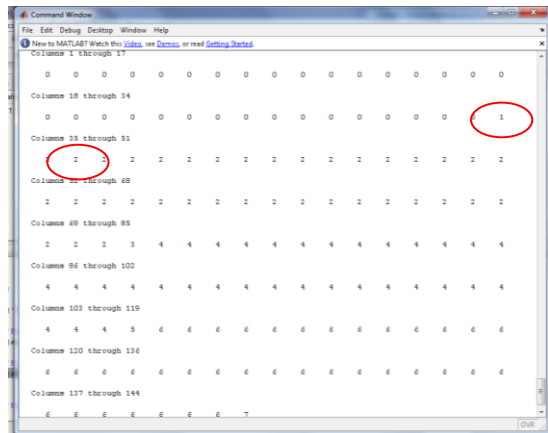
Tahap pengenalan pola dilakukan dengan menganyimpan kode rantai setiap bangun datar

yang diuji yang memiliki ciri khusus pada masing-masing bangun yang nilainya tidak sama dengan kode rantai bangun lainnya, kode rantai acuan yang peneliti gunakan adalah seperti berikut.

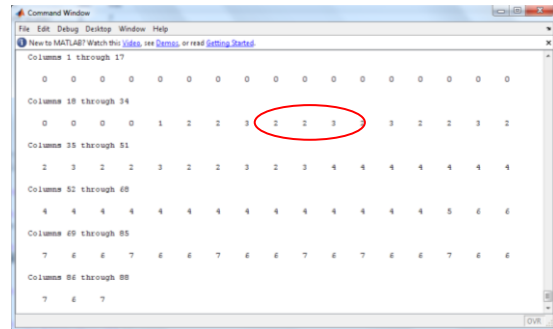
- a. Segitiga = 1 2 1 2 dan 6 7 6 7
- b. Segi empat = 0 1 2 2
- c. Trapesium = 1221
- d. Jajaran genjang = 3 2 2 3



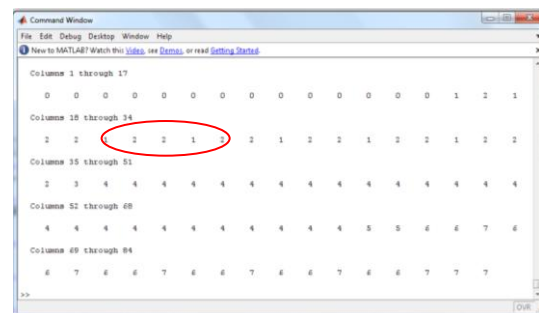
Gambar 24. Kode Rantai Bangun Segitiga (Sumber: data diolah)



Gambar 25. Kode Rantai Bangun Segi empat (Sumber: data diolah)



Gambar 26. Kode Rantai Bangun Jajaran Genjang (Sumber: data diolah)



Gambar 27. Kode Rantai Bangun Trapesium (Sumber: data diolah)

Masing-masing nilai kode rantai bangun datar disimpan sebagai acuan, sehingga jika ada citra baru yang dimasukkan akan dicocokkan dengan kode rantai yang telah disimpan. Apabila kode rantai dari citra masukan memiliki kode rantai yang sama dengan kode rantai acuan maka sistem akan memberikan keluaran berupa deskripsi objek tersebut.

Tahap pengenalan pola pada penelitian ini dilakukan pengujian terhadap 32 macam citra dengan empat macam bentuk bangun datar yaitu segitiga sama sisi, segi empat, jajaran genjang dan trapesium, dan delapan variasi warna objek yaitu merah, kuning, hijau, biru, coklat, hitam, merah muda dan orange. Pada Tabel dibawah dapat dilihat contoh hasil dari pengujian.

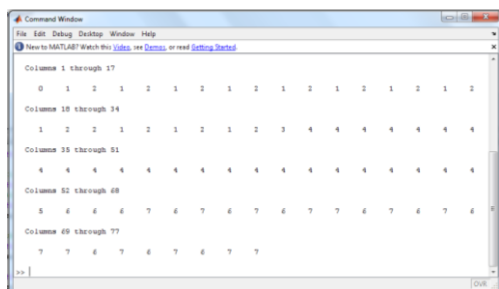


Tabel 3. Hasil Pengujian Pengenalan Bentuk

No	Bentuk Bangun	Nama File	Hasil Pengenalan	Ket.
1		Uji12	Segitiga	Benar
2		Uji4	Segi empat	Benar
3		Uji18	Jajaran Genjang	Benar
4		Uji25	Trapeسيوم	Benar
5		Uji15	Trapeسيوم	Salah
6		Uji6	Segi empat	Benar
7		Uji30	Trapeسيوم	Benar

(Sumber: data diolah)

Dari pengujian yang dilakukan terhadap 32 citra dengan empat macam bangun dan delapan variasi warna objek yang berbeda, citra uji yang tidak dapat teridentifikasi adalah 1. Citra yang tidak teridentifikasi tersebut adalah citra segitiga berwarna hitam yang diberi nama citra uji15. Pada hasil pengujian dapat dilihat bahwa hasil keluaran atau deskripsi tidak sama dengan bentuk bangun aslinya. Kesalahan ini terjadi karena kode rantai citra uji15 memiliki atau mengandung kode rantai yang sama dengan kode rantai trapesium yaitu 1221.



Gambar 28. Kode Rantai Citra Uji15  
(Sumber: data diolah)

**Deskripsi Objek**

Tahap deskripsi objek adalah tahap terakhir pada penelitian ini. Pada tahap ini sistem akan mendeskripsikan hasil pencocokan kode rantai. Apabila kode rantai tersebut cocok dengan kode acuan, maka sistem akan menghasilkan keluaran berupa nama bentuk bangun datar.

**Analisa Pengujian**

Berdasarkan pengujian yang telah dilakukan, diketahui bahwa sistem dapat membaca semua citra uji. Kemudian proses pemotongan citra, perubahan ukuran citra, perubahan warna, pendeteksian tepi, binerisasi, penipisan tepi objek, dan deteksi kode rantai, semuanya dapat berjalan sesuai yang diharapkan. Persentase keberhasilan pada hasil pengenalan bentuk citra menggunakan kode rantai yaitu 96,875%.

**SIMPULAN**

Berdasarkan penelitian yang telah dilakukan, maka diperoleh kesimpulan sebagai berikut.

1. Penelitian ini telah menguraikan implementasi kode rantai untuk pengenalan bentuk bangun datar. Adapun tahapan yang dilakukan antara lain ambil citra, memotong citra, mengubah ukuran citra, mengubah citra masukan yang berupa citra berwarna/RGB menjadi citra abu-abu, mencari tepi objek dengan proses konvolusi menggunakan operator sobel, mengubah citra hasil konvolusi menjadi citra biner, melakukan penipisan tepi objek sehingga memiliki ketebalan satu piksel, mencari nilai kode rantai, melakukan pengenalan bentuk, dan terakhir deskripsi objek.
2. Sistem dapat membaca semua citra uji. Kemudian proses pemotongan citra, perubahan ukuran citra, perubahan warna, pendeteksian tepi, binerisasi, penipisan tepi objek, deteksi kode rantai, pengenalan pola dan deskripsi citra semuanya dapat berjalan sesuai yang diharapkan.
3. Persentase keberhasilan pada sistem pengenalan pola bangun datar menggunakan kode adalah 96,875%.

**DAFTAR PUSTAKA**

- [1] Ahmad, Usman. 2005. *Pengolahan Citra Digital & Teknik Pemrogramannya*. Yogyakarta: Graham Ilmu.
- [2] Gonzales, Rafael C., et al. 2009. *Digital Image Processing using MATLAB*. USA: Gatesmark.
- [3] Kumaseh, Max R., et al. 2013. Segmentasi Citra Digital Ikan menggunakan Metode Thresholding. *Jurnal Ilmiah Sains Vol. 13 No.1; 74-79*.
- [4] Kurnia, M. Riza, et al. 2012. Implementasi Segmentasi Pembuluh Darah Retina pada Citra Fundus Mata menggunakan Tekstur, *Thresholding* dan Operasi Morfologi. *Jurnal Teknik POMITS Vol.1 No.1 ; 1-6*.
- [5] Kristanto, Andri. 2004. *Kecerdasan Buatan*. Yogyakarta: Gaha Ilmu.
- [6] Nain, N., et al. 2006. Morphological Edge Detection and Corner Detection Algorithm using Chain-Enchoding. *IPCV'06*. Department of Computer Engineering India.
- [7] Nuryuliani, et al. 2009. Pengkodean Bentuk Segmen menggunakan Kode Rantai sebagai Dasar Pengenalan Bentuk Karakter Tulisan Tangan secara On-Line. *Seminar on Application and Research in Industri technology*. Jurusan Teknik Mesin dan Industri FT UGM. Pp. A-092 – A-097.
- [8] Prasetyo, Eko. 2011. *Pengolahan Citra Digital dan Aplikasinya menggunakan Matlab*. Yogyakarta: ANDI.
- [9] Pratiwi, M. T. 2013. Aplikasi Kode Rantai untuk Menentukan Keliling dan luas suatu Bangun Datar Dua Dimensi. *Transient Vol. 2 No. 4: 924-929*.
- [10] Putra, Darma. 2010. *Pengolahan Citra Digital*. Yogyakarta: ANDI.