

PENGENALAN KARAKTER MENGGUNAKAN PENDEKATAN ALGORITMA BERBASIS *CHAIN CODE*

Theresia¹, Chris Simon²

Master Computer, Budi Luhur University,
Jl. Raya Ciledug, Jakarta Selatan, Indonesia (12260),
Telp. (021)5853753, Fax.(021)5853752.

¹then_theresia@yahoo.com , ²chris_simon92@yahoo.com

Abstract — The application of computer vision technologies such as OCR (Optical Character Recognition) is an application of pattern recognition technology in the field of artificial intelligence as a reader machine. This study is made to research the application ability to recognize the ASCII Code of the invoice. The processes is using the sample image with Times New Roman, Arial, Calibri typeface that has been scanned, then it will be going through binerization and thinning process using Zhang and Suen Algorithm. If the ASCII Code has passed the thinning process, the chain code and feature calculation will be searched for every character. The distance calculation between the processed image and stored image will be counted using L1-Metric Method. The test will be conducted by taking the character in the processed image sample whether it can be detected correctly based on the stored ASCII Code. After testing is done can be concluded that chain code can be used in ASCII Code recognition on the sample image so a useful system can be made

Index Terms — Computer Vision, Optical Character Recognition, ASCII Code, Chain Code, L1-Metric.

I. PENDAHULUAN

Banyak peralatan elektronik seperti kamera digital, kamera CCTV, video camera, scanner, dan fingerprint reader, yang sering ditemui dan digunakan oleh kebanyakan orang. Alat-alat tersebut nantinya menghasilkan citra digital. Setiap citra yang dihasilkan dapat diolah menggunakan perangkat lunak tertentu yang nantinya akan menghasilkan berbagai macam informasi. Pengolahan citra digital adalah istilah umum untuk berbagai teknik yang keberadaannya untuk memanipulasi dan memodifikasi citra dengan berbagai cara. Efford[1].

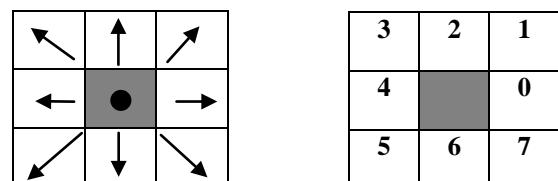
Pengolahan citra merupakan bagian penting yang mendasari berbagai aplikasi nyata, seperti pengenalan pola, penginderaan jarak jauh melalui satelit atau pesawat udara, dan computer vision. Computer Vision banyak sekali di manfaatkan pada bidang multimedia, contohnya Face Detection pada kamera digital, Object Recognition pada konsol game, dan pengenalan karakter atau biasa disebut dengan Optical Character Recognition (OCR).

OCR dapat membaca gambar dan diolah menjadi teks (Image-to-Text). Kita tidak perlu menyalin ulang dokumen yang akan sangat membuat waktu kerja menjadi tidak efisien. Pada penelitian ini akan dibuat OCR dengan menggunakan fitur Kode Rantai (Chain Code). Biasanya fitur Chain Code memang digunakan untuk merepresentasikan kontur seperti mendekripsi tepi ataupun mencari batas daerah. Namun pada penelitian ini akan membuktikan apakah fitur Chain Code dapat digunakan untuk mengenali karakter.

II. METODOLOGI RISET

2.1 Chain Code

Kode rantai adalah suatu kode yang menunjukkan arah pergerakan dari perbatasan luar yang saling menyambung hingga membentuk rantai. Sutoyo, dkk.[5]. Hal ini dapat dilakukan dengan menelusuri sekali lagi piksel-piksel perbatasan dari satu titik hingga kembali ke titik tersebut. Penomoran arah biasanya menggunakan urutan seperti pada Gambar 1.



Gambar 1. Arah Chain Code beserta kodennya

Langkah-langkah untuk mendapatkan kode rantai beserta nilai fiturnya adalah :

1. Mendekripsi dan mengambil arah pergerakan citra sesuai dengan arah chain code.

Pada $x + 1, y - 1$ nilai Chain Code = 1.

Pada $x, y - 1$ nilai Chain Code = 2.

Pada $x - 1, y - 1$ nilai Chain Code = 3.

Pada $x - 1, y$ nilai Chain Code = 4.

Pada $x - 1, y + 1$ nilai Chain Code = 5.

Pada $x, y + 1$ nilai Chain Code = 6.

Pada $x + 1, y + 1$ nilai Chain Code = 7.

Pada $x + 1, y$ nilai Chain Code = 8.

2. Melakukan perhitungan fitur menggunakan rumus :

$$f_i = \frac{\text{(jumlah angka 1)} + \dots + \text{(jumlah angka 8)}}{\text{panjang kode rantai tiap karakter}} \quad (1)$$

2.2 Thinning / Skeleton

Skeleton digunakan untuk representasi dan pengenalan tulisan tangan, pola sidik jari, struktur sel biologis, diagram rangkaian, gambar teknik, rencana jalur robot, dan semacam itu. Salah satu cara untuk mendapatkan skeleton adalah melalui *thinning*. *Thinning* (pengurusan) adalah operasi morfologi yang digunakan untuk memperkecil ukuran geometrik objek dengan hasil akhir berupa skeleton atau rangka. Shih[4].

Metode *thinning* yang digunakan pada penelitian ini adalah metode yang dikembangkan oleh Zhang dan Suen[6]. Metode *thinning* Zhang and Suen menggunakan 2 subiterasi. Sub-iterasi pertama menggerus bagian selatan-timur dari gambar dan sub-iterasi kedua bagian utara-barat. Pada Gambar 2 akan dijelaskan langkah yang digunakan dalam metode *thinning* Zhang and Suen, sebagai berikut :

```

While points are deleted do
    For all pixels p(i,j) do
        if (a) 2 ≤ B(P1) ≤ 6
        (b) A(P1) = 1
        (c) Apply one of the following:
            1. P2 x P4 x P6 = 0 in odd iterations
            2. P2 x P4 x P8 = 0 in even iterations
        (d) Apply one of the following:
            1. P4 x P6 x P8 = 0 in odd iterations
            2. P2 x P6 x P8 = 0 in even iterations
    then
        Delete pixel p(i,j)
    end if
    end for
end while

```

Gambar 2. Langkah metode *thinning* Zhang and Suen

2.3 L1-Metric

L1-Metric melakukan pengukuran jarak antara fitur-fitur yang dimiliki dua buah citra. Dimana jarak kedua buah citra ini yang nantinya akan dipertimbangkan sebagai kemiripan antara dua buah citra. Semakin kecil nilai jarak yang dihasilkan maka kedua citra akan dianggap semakin mirip, semakin besar nilai jarak yang dihasilkan maka kedua citra akan dianggap semakin berbeda. Mulyana[3].

Rumus *L1-Metric* dan dinyatakan dalam persamaan sebagai berikut :

$$d(I, H) = \sum_{l=1}^n |i_l - h_l| \quad (2)$$

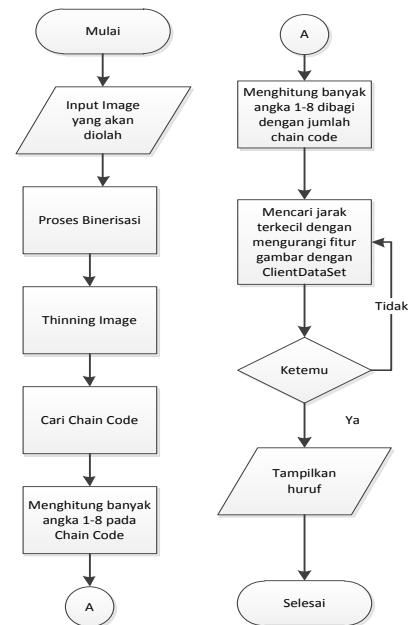
III. PEMBAHASAN

Perangkat keras yang digunakan dalam melakukan proses pengenalan karakter pada penelitian ini akan di jelaskan pada table 1 dibawah :

TABEL 1
 HARDWARE YANG DIGUNAKAN

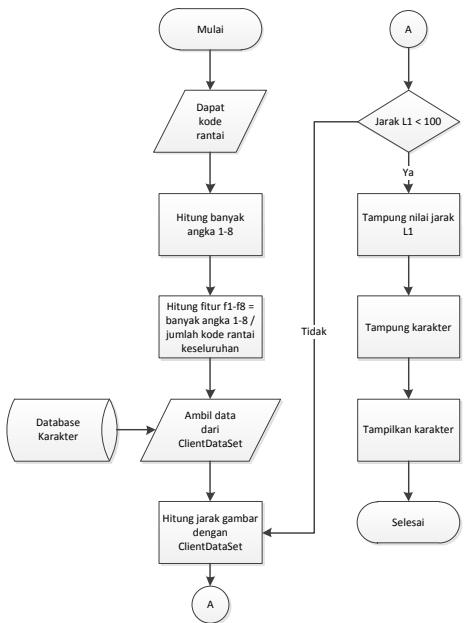
Hardware	Jenis Hadware
Processor	Intel i3 2.53 GHz
Ram	4 GB
Hardisk	50GB <i>available space</i>

Pada Gambar 2 akan dipaparkan *Flow Chart* proses pencarian *Chain Code*.



Gambar 3. *Flow Chart* Pencarian *Chain Code*

Citra akan diubah terlebih dahulu menjadi citra biner, dan akan dibagi kedalam beberapa segmen sesuai dengan banyaknya karakter yang ada pada citra tersebut. Kode rantai akan dicari per masing-masing segmen dan setiap segmen akan ditelusuri arah pergerakan gambarnya dan akan di dapatkan kode rantainya. Jika kode rantai telah didapatkan, maka akan dilakukan perhitungan menggunakan persamaan (1). Hasil dari perhitungan fitur menggunakan persamaan (1) akan digunakan untuk mencari nilai jarak terkecil menggunakan *L1-Metric* dengan cara dihitung dengan jenis-jenis huruf yang sudah disimpan sebelumnya pada *database*. Proses lebih jelas dapat dilihat pada Gambar 4 untuk pencarian *chain code* dan perhitungan jarak.



Gambar 4. Flow Chart Pembagian Chain Code dan Perhitungan Jarak

Perhitungan jarak nantinya akan menampilkan karakter yang mempunyai jarak terkecil setelah dihitung dengan semua karakter pembanding. Pencarian kode rantai dan perhitungan fitur akan berlangsung terus sampai seluruh segmen telah ditelusuri. Sebagai contoh terdapat citra dengan gambar seperti Gambar 5.



Gambar 5. Contoh citra yang akan diproses

Citra diatas menghasilkan kode rantai dan perhitungan banyak angka 1-8 seperti pada Gambar 6.

KODE RANTAI	JML KODE RANTAI	BANYAK ANGKA								HITUNG FITUR							
		1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
03666601444546665561561561	115	9	12	4	13	14	28	7	8	0,078261	0,104348	0,034783	0,113043	0,121779	0,243478	0,06087	0,243478
6655611888888888888812223																	
22323232324668726796726655																	
544404118173787856165666																	
6544548878888																	

Gambar 6. Hasil Penelusuran Kode Rantai Beserta Fiturnya.

Hasil pada Gambar 6 akan dibandingkan dengan kode rantai dan fitur yang tersimpan pada *database*. Gambar 7 akan menunjukkan hasil perhitungan jarak antara citra yang akan diolah dengan citra yang tersimpan.

KARAKTER	HITUNG FITUR PADA CLIENT DATA SET								HITUNG FITUR PADA GAMBAR								LI Metric
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	
A	0,078261	0,090713	0,085179	0,080242	0,079583	0,077127	0,080621	0,078199	0,078261	0,090307	0,084978	0,071364	0,021779	0,243478	0,06087	0,243478	
B	0,113043	0,139073	0,149995	0,099338	0,133340	0,092113	0,134340	0,078261	0,104348	0,050473	0,121779	0,243478	0,06087	0,243478	0,30190528		
A	0,077149	0,048071	0,089269	0,070317	0,071429	0,071978	0,035	0,0602	0,078261	0,051478	0,051783	0,031364	0,021779	0,243478	0,06087	0,243478	
B	0,04790	0,071223	0,080398	0,040247	0,020278	0,020278	0,020278	0,020278	0,020278	0,039473	0,039473	0,039473	0,021779	0,243478	0,06087	0,243478	

Gambar 7. Perhitungan Nilai Jarak Dengan L1-Metric.

Berdasarkan perhitungan jarak pada Gambar 7, didapatkan nilai jarak terkecil yaitu karakter A. Karakter itu akan muncul pada TMemo yang ada pada sistem.

Pengujian dilakukan dengan menggunakan 10 gambar dengan jenis font berbeda dan beberapa karakter yang berbeda juga. Gambar 8 akan menunjukkan pengujian yang sudah dilakukan oleh sistem.

Pada Gambar 8 adalah hasil pengujian sistem yang menggunakan gambar dengan jenis *font* yang telah tersimpan pada *database*. Perhitungan persentase pengujian dilakukan dengan menghitung banyaknya karakter yang ada pada gambar, jumlah karakter benar dan jumlah karakter salah.

No.	Gambar	Banyak Karakter	Benar	Salah	Perse
1	ITIHH 4L4Y BANGEST KAMU	20	13	7	65%
2	CITRA DIGITAL BINER	21	17	3	80,95%
3	inifont timesnew roman	20	8	12	40%
4	PERCOBAAN HURUF VERDANA	20	15	5	75%
5	thisis MsSansSerif font	21	9	12	42,85%
6	PERCOBAAN ANGKA HIRUF	20	7	13	35%
7	THE NETHEREDIA 32199129	20	16	4	80%
8	ANGKANOLU ANGKANOLU	18	9	9	50%
9	1234567890 6543210987 1234567890 6543210987	22	4	18	18,18%
10	OPTICAL CHARACTER 1234	20	15	5	75%
Rata - Rata					56,198%

Gambar 8. Hasil Pengujian Dengan Jenis Font Dengan Jenis Font Yang Sudah Tersimpan Pada Database.

Hasil rata-rata dari sepuluh gambar percobaan yang diuji dengan jenis *font* yang sudah tersimpan pada *database* adalah 56,198%.

No.	Gambar	Banyak Karakter	Benar	Salah	Perse
1	ITIHH 4L4Y BANGEST KAMU	15	9	6	60%
2	GEORGIA VERDANA	21	9	12	42,85
3	inifont timesnew roman	20	9	11	45%
4	PERCOBAAN HURUF VERDANA	20	11	9	55%
5	thisis MsSansSerif font	21	9	14	42,85%
6	PERCOBAAN ANGKA HIRUF	20	10	7	58,82%

7	THE N THERESIA 32100123	20	10	10	50%
8	ANGKANOLO ANGKANOLO	18	9	9	50%
9	0 9 8 7 6 5 4 3 2 1 ROTASI COKLAT	22	4	18	18,18%
10	optical character 1 2 3 4	20	12	8	60%

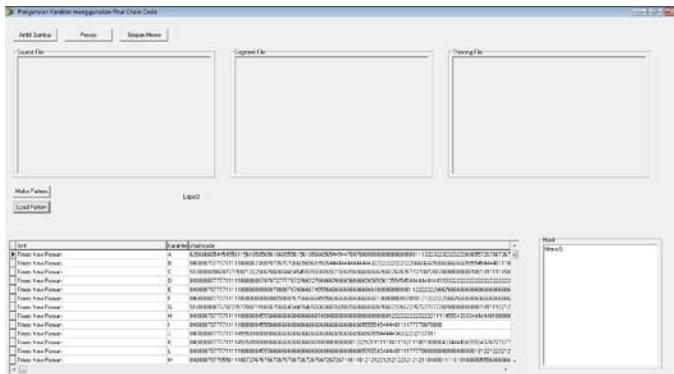
Gambar 9. Hasil Pengujian Sistem Dengan Jenis Font Yang Tidak Tersimpan Pada *Database*.

Hasil rata-rata dari sepuluh gambar percobaan yang diuji dengan jenis *font* yang tidak tersimpan pada *database* adalah 49,77%.

Berdasarkan pada pengujian menggunakan 20 buah gambar yang telah diperlihatkan pada Tabel 4 dan Tabel 5 diperoleh persentase karakter yang benar terbaca adalah 52,945%. Hal ini menunjukkan bahwa fitur *chain code* layak untuk digunakan pada pengenalan karakter.

IV. IMPLEMENTASI SISTEM

Berikut akan dijelaskan tahap-tahap penggunaan sistem yang telah dibangun, dimana pada Gambar 10 menunjukkan tampilan halaman utama sistem.



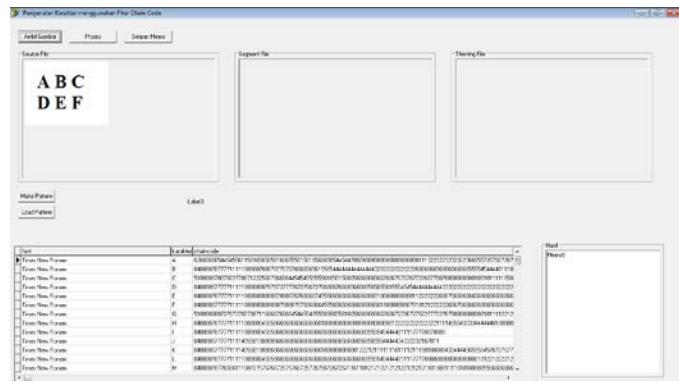
Gambar 10. Tampilan utama sistem.

Pada Gambar 11 menjelaskan tahap melakukan import gambar yang akan dikenali, sebagai berikut :



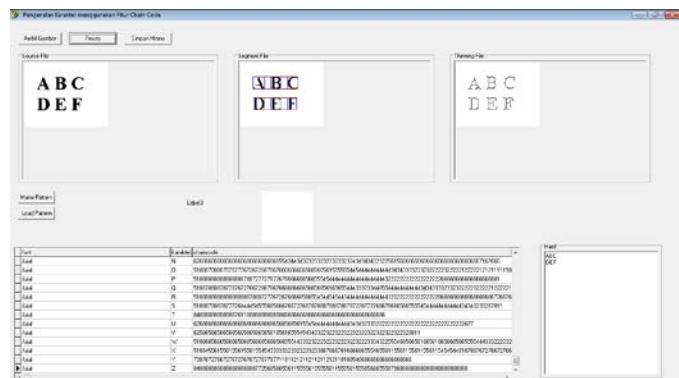
Gambar 11. Import gambar didalam sistem.

Pada Gambar 12 menjelaskan tahap proses pada saat melakukan recognition untuk mengenali kode ASCII pada gambar yang diimport, sebagai berikut :



Gambar 12. Proses recognition sistem.

Pada Gambar 13 merupakan tahap akhir dalam penggunaan sistem, dimana sistem mengeluarkan hasil *output* antara lain tahap segmentasi, tahap *thinning* dan *output* kode ASCII yang telah dibandingkan dengan *database* menggunakan *L1-Metic*, sebagai berikut :



Gambar 13. Hasil recognition sistem.

V. KESIMPULAN

5.1 Kesimpulan

Setelah dilakukan implementasi fitur *chain code* pada pengenalan karakter, dapat diambil kesimpulan sebagai berikut :

1. Metode *chain code* layak diterapkan pada pengenalan karakter karena memiliki persentase keberhasilan 52,945% sesuai dengan hasil pengujian.
 2. Sistem dapat mendeteksi gambar yang terdapat campuran huruf besar, huruf kecil dan angka.
 3. Pembacaan karakter berdekatan kurang baik.
 4. Belum dapat membedakan antara huruf besar dan huruf kecil pada gambar meskipun huruf yang telah diproses benar, dikarenakan terdapat kesamaan ciri pada tiap karakter.
 5. Sistem belum dapat membaca gambar berupa foto.
 6. Sistem dapat membaca gambar yang dirotasi, namun hasilnya tidak sesuai dengan masukkan.

7. Sistem belum dapat membaca dengan baik jika terdapat karakter yang memiliki kecacatan gambar, seperti huruf 'O' yang hilang bagian ujungnya bisa benar terbaca huruf 'O' ataupun huruf yang memiliki ciri serupa, tergantung dari tingkat kecacatan huruf.

5.2 Saran

Saran yang sekiranya berguna untuk perkembangan lebih lanjut dari penelitian ini adalah :

1. Dapat membaca sekumpulan karakter yang sama.
2. Dapat membaca spasi pada gambar dan karakter dengan jarak spasi yang dekat.
3. Dapat membaca gambar yang didalamnya terdapat tulisan tangan.
4. Terdapat lebih banyak jenis *font* yang disimpan dalam *database*.
5. Terdapat fitur untuk rotasi gambar, jika posisi gambar belum sesuai, serta fitur untuk menghilangkan *noise*, ataupun mencerahkan gambar.

DAFTAR PUSTAKA

- [1] Efford, N., "Digital Image Processing a Practical Introduction Using Java", Essex : Pearson Education Limited, 2000.
- [2] Kadir, Abdur,"Dasar Pengolahan Citra dengan Delphi", Yogyakarta : Andi, 2013.
- [3] Mulyana, T.M.S., "Pengenalan Gerakan Tangan Secara Optis", Jakarta, 2013.
- [4] Shih, F.Y., "Image Processing and Mathematical Morphology", New York : CRC Press, 2009.
- [5] Sutoyo, T., S.Si., dkk. "Teori Pengolahan Citra Digital", Yogyakarta : Andi, 2009.
- [6] T.Y. Zhang and C.Y. Suen (1984). A Fast Parallel Algorithm for Thinning Digital Patterns. *Communication of the ACM*, 27(3), pp 236-239. Available: www.prima.inrialpes.fr/perso/Tran/Draft/gateway.cfm.pdf



Theresia, dilahirkan di Jakarta pada tanggal 4 Desember 1992. Merupakan anak kedua dari dua bersaudara. Pada tahun 2007 menempuh pendidikan menengah (SMA) di sekolah Vianney, Jakarta Barat, kemudian pada tahun 2010 menempuh pendidikan Sarjana (S1) di Universitas Bunda Mulia, Jakarta Utara dengan menempuh jurusan Teknik Informatika. Setelah lulus dan mendapatkan gelar sarjananya, kemudian mencari pekerjaan sebagai developer java hingga saat ini.



Chris Simon, dilahirkan di Jakarta pada tanggal 26 Juli 1992. Merupakan anak pertama dari tiga bersaudara. Pada tahun 2007 menempuh pendidikan menengah (SMA) di sekolah Vianney, Jakarta Barat, kemudian pada tahun 2010 menempuh pendidikan Sarjana (S1) di Universitas Bunda Mulia, Jakarta Utara dengan menempuh jurusan Teknik Informatika (TI) dan pada tahun 2014 mengambil pendidikan lanjut Magister (S2) di Universitas Budi Luhur, Jakarta Selatan dengan menempuh jurusan yang sama yaitu Teknik Informatika (TI) hingga saat ini.