

Pengenalan Karakter Tulisan Tangan Jawi Menggunakan Metode *New Relative Context* dan SVM

Rizal Fikri¹, Fitri Arnia², Rusdha Muharar³

Abstract— Dot is an important attribute in character recognition. Similarly in Jawi characters, a dot becomes a special characteristic that distinguish different characters with the same basic shape. Most of feature extraction methods only recognize the characters based on their basic shape and ignore the dots, such as *Relative Context* (RC). RC classifies characters with the same basic shape into a group. Therefore, the result recognition of RC is not individual characters, but the name of group character. To identify individual character, a new method for RC enhancement is introduced. The method is called *New Relative Context* (NRC). NRC works by separating characters into some areas. The wider area is defined as the basic shape, while other areas are defined as dot attribute. In this paper *Support Vector Machine* (SVM) is used to classify eleven sets of isolated Jawi characters. Eight sets of character images are used in the training phase, while in the testing phase three sets of images are used. The recognition rate of this method achieves 80%.

Intisari— Tanda titik pada sebuah karakter merupakan unsur penting dalam pengenalan sebuah karakter. Tak terkecuali pada karakter Jawi, tanda titik menjadi sebuah ciri khusus untuk membedakan beberapa karakter yang mempunyai bentuk dasar yang sama. Sebagian metode ekstraksi fitur mengenali karakter hanya pada bentuk dasarnya dengan menghilangkan unsur tanda titiknya, seperti *Relative Context* (RC). RC mengelompokkan bentuk dasar yang sama ke dalam sebuah kelompok sehingga hasil pengenalan metode ekstraksi fitur RC adalah nama kelompok dan bukan karakter asli yang diinginkan. Untuk menyempurnakan metode RC agar dapat mengenali karakter asli lengkap dengan titiknya, pada makalah ini diperkenalkan sebuah metode ekstraksi fitur baru yang merupakan modifikasi RC yang selanjutnya disebut *New Relative Context* (NRC). NRC bekerja dengan memisahkan bagian aksara ke dalam beberapa area. Area yang paling luas didefinisikan sebagai bagian utama sedangkan area yang lainnya didefinisikan sebagai bagian titik. Karakter diklasifikasi dengan menggunakan *Support Vector Machine* (SVM). Pengujian dilakukan pada sebelas set karakter dari semua aksara Jawi, dan hasil pengujian dengan delapan set citra *training* dan tiga set citra *testing* memberikan performa keberhasilan pengenalan sampai dengan 80%.

Kata Kunci— pengenalan karakter, karakter Jawi, NRC, SVM.

¹ Mahasiswa, Magister Teknik Elektro Program Pasca Sarjana Universitas Syiah Kuala, Jl. Tgk. Syech Abdul Rauf No. 7 Darussalam - Banda Aceh, INDONESIA, (e-mail: rizalfikri@gmail.com)

^{2,3} Dosen, Magister Teknik Elektro Program Pasca Sarjana Universitas Syiah Kuala, Jl. Tgk. Syech Abdul Rauf No. 7 Darussalam - Banda Aceh, INDONESIA, (e-mail: ²f.arnia@unsyiah.ac.id, ³r.muhaarar@unsyiah.ac.id)

I. PENDAHULUAN

Tulisan Jawi ialah tulisan yang digunakan oleh bangsa Melayu yang karakter-karakternya diambil dari karakter Arab. Perkataan Jawi berasal dari bahasa Arab dan disebut Yawi di Pattani, yaitu abjad Arab yang disesuaikan untuk menulis bahasa Melayu. Perkataan ini adalah kata sifat dari kata Arab “*Jawah*” dan apabila dijadikan kata sifat “*Jawah*” menjadi “*Jawi*”. Aksara Jawah atau Jawi tidak hanya digunakan di Indonesia saja, namun juga di seluruh Asia Tenggara [1].

Tulisan Jawi merupakan salah satu tulisan tangan yang termasuk ke dalam golongan tulisan tangan kursif, yaitu tulisan tangan yang menyambung antara satu karakter dengan karakter lainnya. Akhir-akhir ini banyak penelitian telah dikhususkan untuk pengenalan karakter tulisan tangan kursif baik secara *online* maupun *offline*, seperti kursif Latin, Arab, Urdu, dan Farsi [2]. Alasan utama bidang penelitian ini masih terbuka untuk diteliti adalah karena karakteristik teks kursif yang berbeda dan lebih kompleks dibandingkan dengan jenis tulisan non-kursif. Kompleksitas ini disebabkan oleh fakta bahwa pada tulisan tangan kursif karakter yang sama sering ditulis berbeda, dan karakter yang berbeda sering ditulis sangat mirip antara satu sama lainnya [3]. Selain alasan di atas, perbedaan teknik ekstraksi fitur juga menyebabkan tingkat keberhasilan yang berbeda sehingga menjadi sebuah ruang penelitian baru dengan menggunakan ekstraksi fitur yang berbeda [4].

Sebuah karakter memiliki ciri khusus yang menjadi sebuah tanda untuk mengenali karakter tersebut. Ciri tersebut dapat berupa berbagai macam parameter sesuai dengan jenis fitur yang digunakan. Ekstraksi fitur adalah proses mendapatkan nilai penting dan unik untuk membedakan satu objek dengan objek yang lain [5]. Ekstraksi fitur yang efektif dan efisien merupakan faktor penting dalam mencapai performa yang bagus dalam pengenalan karakter [6]. Selain pemilihan metode ekstraksi fitur, pada pengenalan karakter dengan tingkat kerumitan yang tinggi perlu dipertimbangkan penggabungan beberapa metode ekstraksi fitur untuk mencapai performa pengenalan yang lebih optimal [7]. Banyak jenis metode ekstraksi fitur yang telah digunakan pada pengenalan karakter Arab dan Jawi, salah satunya adalah *Relative Context* (RC) [8]. RC bekerja dengan menghilangkan tanda titik dari sebuah karakter dan mengelompokkan bentuk dasar karakter yang sama ke dalam sebuah kelompok sehingga hasil pengenalan metode ekstraksi fitur RC adalah nama kelompok dan bukan karakter asli yang diinginkan.

Karakter Jawi dikenali bukan hanya dari bentuk dasarnya melainkan juga dikenali dari tanda titik, yaitu letak titik dan banyaknya titik. Hal ini menjadi sebuah permasalahan tersendiri bagi RC dalam mengenali huruf Jawi karena RC

tidak mampu mengenali atribut titik dari sebuah karakter. Untuk mengenali ketiga unsur tersebut maka diperlukan sebuah metode ekstraksi fitur yang bisa membedakan antara bentuk dasar, banyak titik, dan letak titik dari karakter tersebut. Pada makalah ini diperkenalkan sebuah metode ekstraksi fitur baru yang merupakan modifikasi dari RC yang dinamakan *New Relative Context* (NRC). NRC mampu mendeteksi bentuk dasar dari sebuah karakter dan juga mampu memprediksi banyaknya titik dan posisi titik dari sebuah karakter yang sebelumnya tidak bisa dilakukan oleh metode RC. Teknik pengklasifikasian yang digunakan pada makalah ini adalah metode *Support Vector Machine* (SVM) [9].

Makalah ini terdiri atas beberapa bagian. Bagian kedua menjelaskan teknik pengambilan nilai fitur dengan metode NRC. Bagian ketiga memaparkan tentang cara penelitian dilakukan, dan pada bagian keempat dijelaskan hasil dan analisis pengujian dari metode ekstraksi fitur NRC.

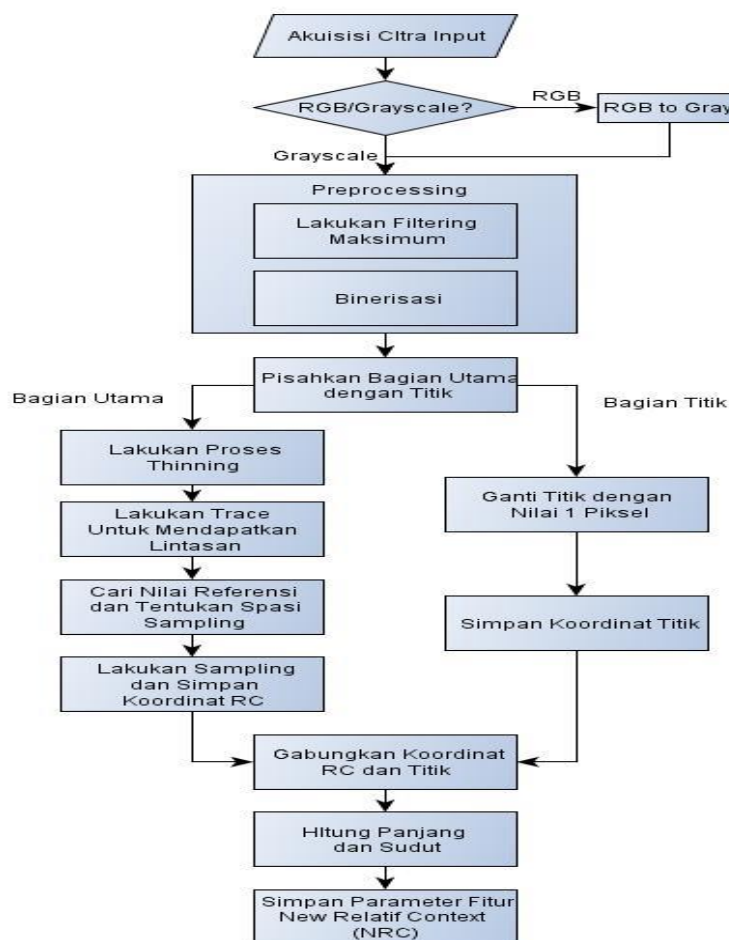
II. NEW RELATIVE CONTEXT (NRC)

New Relative Context (NRC) merupakan metode ekstraksi fitur hasil modifikasi dari RC. Metode NRC juga menggunakan perbandingan panjang antara koordinat-koordinat RC dengan titik referensi dan juga nilai sudut yang terbentuk dari koordinat-koordinat tersebut dengan titik referensi terhadap *axis horizontal*. Titik referensi pada NRC

ditentukan dengan membagi dua panjang lintasan *Spline* (*sampling line*). Setelah koordinat RC didapat pada NRC selanjutnya ditambahkan metode untuk mengenali titik dan banyaknya titik. Teknik ekstraksi fitur untuk mendapatkan nilai fitur NRC secara lengkap ditunjukkan oleh algoritme pada Gbr. 1 dan dijelaskan pada bagian A sampai dengan K.

A. Tahapan Preprocessing

Preprocessing merupakan teknik menormalkan atau untuk menghapus variasi yang lain (*noise*) dari citra awal yang telah dipindai menggunakan alat pemindai dengan tujuan mempermudah proses identifikasi. Ada beberapa faktor yang mempengaruhi kualitas pengenalan sebuah karakter di antaranya adalah kualitas *scanner*, resolusi *scanner*, jenis dokumen yang dicetak (*printer laser* atau fotokopi), kualitas kertas, *font* yang digunakan dalam teks, kompleksitas tata bahasa, dan kamus yang digunakan. Untuk mengatasi hal-hal tersebut, diperlukan tahapan-tahapan *preprocessing*. Penggunaan algoritme, peningkatan akurasi gambar, penghapusan *noise*, gambar *thresholding*, deteksi/koreksi, segmentasi halaman, segmentasi karakter, normalisasi karakter merupakan tahap penting dalam *preprocessing* untuk mencapai sebuah hasil yang efektif. Hasil yang diinginkan dari *preprocessing* khususnya dalam pengenalan karakter adalah teks gambar yang berisi biner saja [10].



Gbr. 1 Teknik ekstraksi fitur NRC.

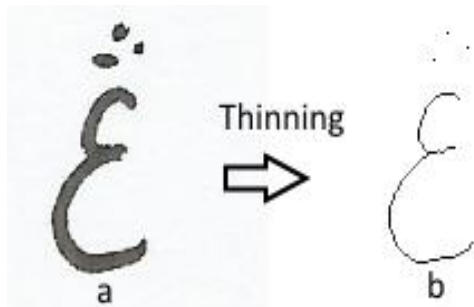
Pada pengenalan huruf Jawi dengan menggunakan metode NRC digunakan beberapa jenis operasi *preprocessing*, di antaranya adalah *filtering* maksimum dan binerisasi.

Filtering maksimum digunakan pada pengenalan karakter huruf Jawi untuk menghilangkan derau akibat luapan tinta yang mengakibatkan terjadinya percabangan ataupun perubahan bentuk dasar sebuah karakter. Dengan latar belakang bernilai maksimum dan bentuk karakter bernilai minimum, maka jika proses *filtering* maksimum dilakukan secara otomatis rembesan tinta yang terjadi akan hilang atau tergantikan dengan nilai maksimum sehingga bentuk dasarnya tetap terjaga. Proses *Filtering* dilakukan dengan teknik konvolusi yaitu perkalian antara elemen matriks citra dengan elemen matriks penapis (*kernel*) yang dilakukan dengan menggeser jendela matriks tapis piksel per piksel. Setiap hasil perkalian kemudian disimpan dalam matriks baru [11].

Binerisasi dilakukan untuk mendapatkan citra yang hanya mempunyai dua degradasi warna. Degradasi pertama bernilai 1 sebagai bentuk dasar karakter, dan degradasi bernilai 0 sebagai latar belakang dari karakter tersebut

B. Operasi Thinning

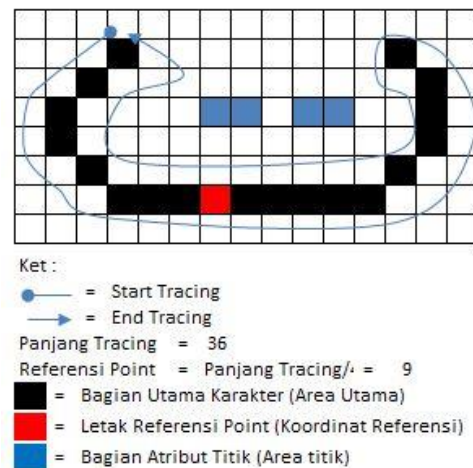
Operasi *thinning* digunakan untuk membuat ketebalan tulisan dari karakter menjadi satu piksel tanpa mengubah bentuk dasar dari sebuah karakter. Operasi ini diperlukan untuk mempermudah proses pengenalan karakter pada tahapan selanjutnya. Gbr. 2(a) menunjukkan *input* aksara Jawi “nga”, sedangkan Gbr. 2(b) menunjukkan operasi aksara Jawi “nga” setelah mengalami operasi *thinning*.



Gbr. 2 Operasi *thinning*.

C. Memisahkan Bagian Utama dan Titik

Sebagian karakter Jawi mempunyai atribut titik sebagai pembeda antara beberapa karakter dengan bentuk dasar yang sama. Untuk melakukan proses pengenalan karakter pada aksara yang mempunyai atribut titik perlu dipisahkan antara bentuk dasar karakter dan atribut titiknya. Pada bagian ini, pemisahan dilakukan dengan melakukan pencarian luas area dari bentuk-bentuk citra tersebut, sebagai contoh, pada karakter “ت” terdapat tiga buah area yaitu area pertama adalah area yang jumlah pikselnya lebih dari 40, yang selanjutnya didefinisikan sebagai bagian utama, kemudian untuk area kedua dan ketiga yang jumlah pikselnya kurang dari 40, didefinisikan sebagai bagian titik. Pada Gbr. 3 ditunjukkan teknik pemisahan area utama dan area titik pada aksara “ت”. Area utama ditunjukkan oleh bagian yang berwarna hitam, sedangkan area titik ditunjukkan oleh area yang berwarna biru.



Gbr. 3 Pemisahan area dan *tracing boundary*.

D. Mengganti Bagian Titik Menjadi Satu Piksel

Setelah pemisahan dilakukan, selanjutnya dilakukan perhitungan nilai pusat massa pada masing-masing area yang didefinisikan sebagai titik. Setelah nilai pusat massa didapat kemudian dihilangkan semua nilai piksel yang bernilai 1 pada area titik tersebut, dan selanjutnya nilai piksel pada pusat massa dari masing-masing area titik diganti dengan nilai 1. Proses ini dilakukan dengan tujuan untuk memastikan bahwa satu tanda titik hanya diwakili oleh satu *end-point*.

E. Menyimpan Koordinat Titik

Setelah semua area titik didapatkan dan diubah menjadi satu piksel, selanjutnya yang perlu dilakukan adalah menyimpan koordinat-koordinat titik tersebut karena koordinat-koordinat titik tersebut merupakan bagian dari koordinat NRC.

F. Melakukan Trace Boundary

Trace boundary dilakukan untuk mendapatkan setiap koordinat lintasan dari bentuk karakter. *Trace boundary* bekerja dengan dimulai dari *end-point* pada kolom terkecil dari sebuah citra, dan akan berhenti pada koordinat pertama dimulai. Pada Gbr. 3 dapat dilihat proses *tracing boundary* seperti yang ditunjukkan oleh lintasan garis biru. Nilai koordinat dari *trace boundary* disimpan dalam matriks dua kolom dan N baris. N merupakan banyaknya koordinat hasil dari *tracing*. Kolom pertama matriks tersebut merupakan koordinat baris dari *tracing* dan kolom kedua merupakan koordinat kolom dari *tracing*.

G. Menentukan Titik Referensi

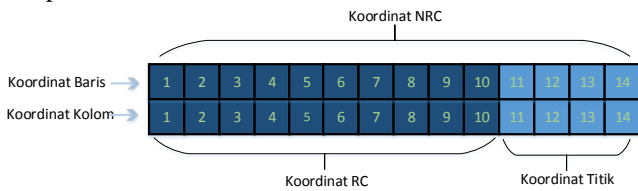
Titik referensi didapatkan dengan membagi empat panjang lintasan dari *tracing* yang dilakukan. Proses ini dilakukan untuk mendapatkan referensi berada tepat di tengah-tengah dari lintasan *tracing*. Tahapan selanjutnya adalah menentukan banyaknya *sampling*. Jika banyaknya koordinat yang diinginkan adalah sepuluh, maka *sampling* dilakukan sebanyak lima kali ke kiri dan lima kali ke kanan dari titik referensi dengan jaraknya *sampling* adalah banyaknya N baris koordinat hasil *trace boundary* dibagi 20.

H. Menyimpan Koordinat Sampling

Setelah frekuensi *sampling* ditentukan, tahapan selanjutnya adalah melakukan *sampling* pada lintasan *tracing* dan menyimpan koordinat hasil *sampling* tersebut yang merupakan koordinat RC pada bagian utama dari sebuah karakter.

I. Menggabungkan Koordinat Utama dan Titik

Pada bagian ini dilakukan penggabungan koordinat *sampling* dari bagian utama karakter dan koordinat dari atribut titik menjadi satu matriks 2x14. Kolom 1 sampai dengan 10 diisi dengan koordinat *sampling* dari bentuk dasar karakter, selanjutnya kolom ke 11 sampai dengan 14 diisi dengan koordinat dari atribut titik. Bentuk akhir koordinat NRC ditunjukkan oleh Gbr. 4. Jika karakter yang diuji tidak mempunyai atribut titik maka nilai dari kolom 11 sampai dengan 14 diisi dengan nilai 0. Bentuk terakhir hasil penggabungan NRC adalah sepuluh kolom pertama sebagai koordinat dari bagian utama dan empat kolom selanjutnya merupakan koordinat titik.



Gbr. 4 Susunan koordinat *New Relative Context* (RC).

J. Menghitung Panjang dan Sudut Antara Koordinat NRC dan Koordinat Referensi

Dalam tahapan ini dilakukan perhitungan panjang dan sudut secara berurutan dari koordinat NRC pertama sampai terakhir. Panjang yang dihitung adalah panjang antara tiap-tiap koordinat NRC dengan koordinat referensi dan juga sudut yang terbentuk dari tiap-tiap garis tersebut terhadap *axis horizontal*. Perhitungan panjang $P(i)$ dan sudut $\beta(i)$ dapat dilakukan dengan menggunakan (1) dan (2).

$$P(i) = \sqrt{\{X - X(i)\}^2 + \{Y - Y(i)\}^2} \tag{1}$$

$$\beta(i) = \tan^{-1}\{(X - X(i))/(Y - Y(i))\} \tag{2}$$

dengan $X(i)$ menyatakan koordinat baris, $Y(i)$ menyatakan koordinat kolom, sedangkan X dan Y masing –masing adalah koordinat referensi baris dan kolom. Setelah semua nilai panjang dan sudut didapat, tahap selanjutnya adalah membagi tiap-tiap fitur panjang yang didapat dengan panjang maksimum dari fitur itu sendiri sehingga nilai $P(i)$ yang didapat bernilai maksimum 1.

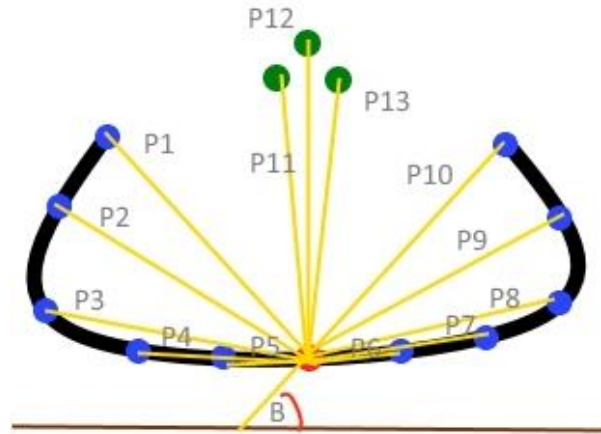
K. Menyusun dan Menyimpan Fitur

Setelah fitur panjang dan fitur sudut didapat, selanjutnya fitur tersebut disusun berpasangan sesuai dengan urutan (3).

$$Fitur\ NRC = \{P(1), \beta(1), P(2), \beta(2), \dots, P(n), \beta(n)\} \tag{3}$$

Selanjutnya nilai fitur tersebut disimpan untuk tiap-tiap karakter. Baik karakter *training* maupun karakter *testing* akan

melalui proses yang sama untuk mendapatkan nilai fitur tersebut.



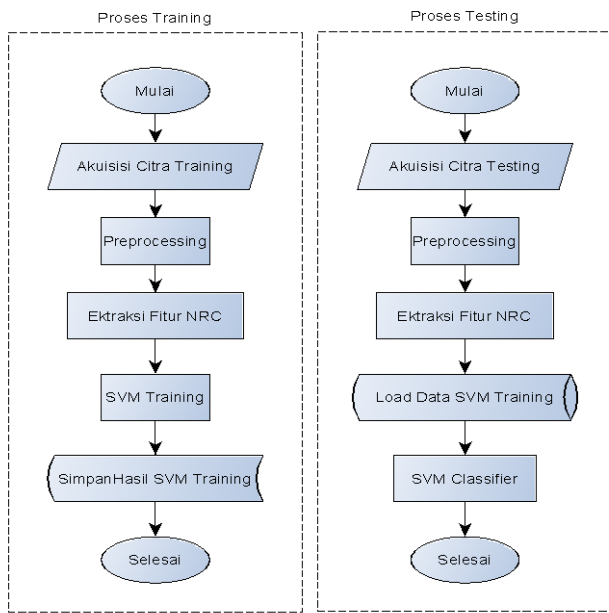
Gbr. 5 Representasi huruf “nya” dalam *New Relative Context* (RC).

Pada Gbr. 5 dapat dilihat bagaimana fitur dari metode NRC didapat. Dari gambar tersebut juga dapat dilihat bahwa $P1$ sampai dengan $P13$ merupakan panjang yang didapat antara koordinat NRC dengan koordinat referensi. Simbol β yang terlihat pada Gbr. 5 merupakan sudut yang terbentuk antara garis panjang tersebut terhadap *axis horizontal*. Sudut ini akan diambil sebanyak koordinat NRC. Secara keseluruhan panjang fitur NRC adalah pasangan dari panjang dan sudut yang terbentuk dari koordinat-koordinat NRC.

III. METODOLOGI

Penelitian pengenalan huruf Jawi dengan memperkenalkan metode ekstraksi baru NRC dilakukan dalam beberapa tahap. Tahap awal dari penelitian adalah mempersiapkan bahan penelitian yaitu bentuk-bentuk tulisan tangan huruf Jawi yang terpisah (*isolated*) sebanyak sebelas sampel untuk masing-masing karakter. Dari sebelas sampel bentuk tulisan tangan tersebut delapan di antaranya digunakan sebagai masukan citra *training*, sedangkan sisanya sebanyak tiga sampel digunakan sebagai citra *testing*. Tahap selanjutnya adalah membangun aplikasi simulasi yang dibuat dengan bahasa pemrograman Matlab. Aplikasi dibangun dengan mengikuti prosedur sesuai dengan bentuk *flowchart* pada Gbr. 6. Aplikasi dibuat sebanyak dua buah, yaitu aplikasi untuk simulasi citra *training* dan aplikasi untuk simulasi citra *testing*.

Pada proses simulasi, yang pertama kali harus dilakukan adalah proses *training*. Tujuan dari proses ini adalah untuk mendapatkan model hasil klasifikasi dengan SVM yang selanjutnya disimpan dalam *database*. Model klasifikasi yang dibangun pada simulasi ini adalah metode SVM *One-Again-One* (OAO) dengan banyaknya model adalah $k(k - 1)/2$. Nilai k merupakan banyaknya karakter terpisah huruf Jawi yaitu 35, sehingga model yang terbentuk menjadi 595 model. Setelah proses *training* selesai, selanjutnya baru dijalankan proses *testing* untuk tiap-tiap karakter yang diuji, sehingga dihasilkan hasil klasifikasi dari masing-masing karakter tersebut. Hasil klasifikasi inilah yang kemudian menjadi keluaran untuk mengukur tingkat keberhasilan sistem dan metode yang dibangun khususnya metode NRC.



Gbr. 6 Tahapan proses simulasi.

IV. HASIL DAN PEMBAHASAN

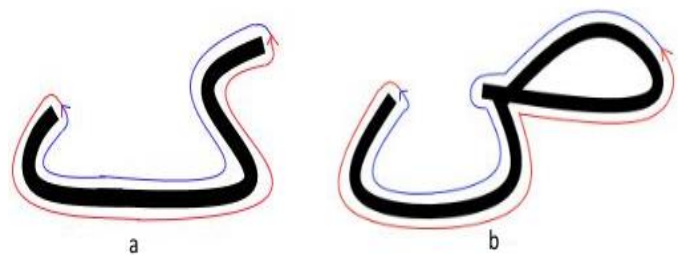
Penggunaan SVM *kernel* akan memberikan hasil yang berbeda dalam tingkat keberhasilan pengklasifikasian. Untuk memperoleh *kernel* yang memberikan hasil yang terbaik perlu dilakukan pengujian untuk setiap *kernel* yang ada. Tingkat keberhasilan dari setiap *kernel* yang diuji dapat dilihat pada Tabel I.

Metode NRC yang diuji dan digunakan pada sistem ini menggunakan sepuluh koordinat RC untuk bentuk dasarnya, sedangkan untuk koordinat titik disediakan empat koordinat. Apabila karakter yang di-*training* maupun di-*testing* tidak memiliki titik, maka nilai koordinat ini adalah (0,0) yang secara otomatis jarak koordinat (0,0) tersebut dengan referensi *point* dan juga besarnya sudut diberikan nilai menjadi 0. Dengan menggunakan sepuluh koordinat RC dan empat koordinat titik maka panjang fitur NRC yang terbentuk adalah 28, yang diletakkan secara berpasangan dimulai dengan panjang kemudian sudut.

Bahan pengujian dalam makalah ini didapat dari hasil akuisisi citra aksara Jawi terpisah sebanyak sebelas set sampel dengan masing-masing set terdiri atas 35 aksara. Dari sebelas sampel, delapan set di antaranya dijadikan sebagai citra *training* sedangkan tiga set sisanya dijadikan citra *testing*.

Hasil pengujian dari sebelas set citra aksara Jawi ditunjukkan oleh Tabel I. Dari tabel tersebut dapat dilihat bahwa metode NRC mempunyai performa yang baik dengan tingkat pengenalan sampai dengan 80%. Performa ini didapat dari uji coba dengan SVM *kernel Linear* dan RBF, sedangkan untuk *kernel* yang lain menghasilkan hasil yang kurang memuaskan, dengan tingkat keberhasilan minimum 45,71% pada penggunaan *kernel radial*.

NRC bekerja dengan melakukan *tracing boundary* dari bentuk utama karakter tersebut, namun pada pengambilan koordinatnya hanya diambil dari setengah lintasan dari *tracing boundary*, seperti yang terlihat pada Gbr. 7(a). Pengambilan koordinat dari setengah *tracing boundary* akan efektif jika karakter tersebut tidak mempunyai percabangan dalam penulisannya. Keefektifan ini disebabkan karena posisi koordinat akan terwakili untuk keseluruhan bagian karakter. Pada Gbr. 7(a) dapat terlihat bahwa pengambilan koordinat sepanjang lintasan garis biru dan lintasan garis merah akan menunjukkan posisi koordinat yang sama, sehingga pengambilan koordinat dari setengah *tracing* akan mewakili seluruh bagian karakter.



Gbr. 7(a) *Tracing boundary* tanpa percabangan, (b) *Tracing boundary* dengan percabangan.

Berbeda untuk karakter yang mempunyai percabangan, teknik pengambilan koordinat ini tidak begitu efektif karena akan ada beberapa ciri khusus sebuah karakter yang hilang. Pada Gbr. 7(b) dapat dilihat bahwa jika *sampling* hanya dilakukan pada lintasan garis merah maka sebagian dari karakter yang dilewati lintasan garis biru tidak digunakan sebagai sebuah ciri khusus oleh metode NRC. Begitu juga dengan keadaan sebaliknya. Adanya beberapa bagian yang tidak diambil koordinatnya pada karakter yang mempunyai percabangan menyebabkan performa pengenalan menurun karena banyaknya kesalahan pengenalan terjadi pada aksara yang mempunyai percabangan tersebut.

TABEL I
HASIL PENGUJIAN METODE EKSTAKSI FITUR NRC

No.	Banyaknya Citra Masukan		SVM <i>Kernel</i>	Hasil Pengenalan Benar		Hasil Pengenalan Salah	
	<i>Training</i>	<i>Testing</i>		Angka	%	Angka	%
1	280	105	<i>Linear</i>	84	80	21	20
2	280	105	RBF ($C=100, rbf\ sig =24$)	84	80	21	20
3	280	105	<i>Polynomial</i>	75	71.43	30	28.57
4	280	105	<i>Quadratic</i>	74	74.47	31	29.53
5	280	105	<i>Radial</i>	48	45.71	57	54.29

V. KESIMPULAN

Setelah dilakukan pengujian dalam penelitian, dapat disimpulkan bahwa metode ekstraksi fitur NRC yang dikembangkan dari metode RC bekerja cukup baik dalam proses pengklasifikasian huruf Jawi lengkap dengan atribut titiknya dengan tingkat keberhasilan 80%. SVM *kernel* yang menghasilkan performa yang paling bagus dengan menggunakan metode ekstraksi fitur NRC adalah dengan menggunakan SVM tanpa *kernel* yaitu SVM *linear* dan SVM dengan menggunakan *kernel* RBF dengan nilai $C=100$ dan $rbf\ sigma=24$.

REFERENSI

- [1] A. J. Borham, "Tulisan Jawi : Tulisan Serantau," *Seminar Tulisan Jawi dan Teknologi Peringkat Kebangsaan.*, Universiti Malaysia Pahang - Pahang, 2012.
- [2] M. F. Nasrudin and M. Petrou, "Offline Handwritten Jawi Recognition using the Trace Transform," *Proc. International Conference on Pattern Analysis and Intelligent Robotics.*, Vol 1. pp. 87-91, June 28-29, 2011.
- [3] M. F. Nasrudin, K. Omar, M. S. Zakaria, and L. C. Yeun, "Handwritten Cursive Jawi Character Recognition: A Survey," *Proc. Fifth International Conference on Computer Graphics, imaging and Visualization.*, pp. 247-256, 2008.
- [4] A. Marwa, Z. Kamel, Z. Salah, and G. Khaled, "Arabic Character Recognition Based M-SVM: Review," *Proc. Second International Conference Advanced Machine Learning Technologies and Applications.*, pp. 18-25, November 28-30, 2014.
- [5] M. Kusban, "Verifikasi dan Identifikasi Telapak Tangan dengan Kernel Gabor", *JNTETI*, Vol. 4, No. 2, Mei 2015.
- [6] A. D. Trier, A. K. Jain, and T. Taxt, "Feature Extraction Methods for Character Recognition - A Survey", *Pattern Recognition*, vol. 29, pp. 641-662, 1996.
- [7] M. A. Abuzaraiza, A. M. Zeki, and A. M. Zeki, "Feature Extraction Techniques of Online Handwriting Arabic Text Recognition", *Proc. Fifth International Conference on Information and Communication Technology for the Muslim World.*, pp.1-7, March 26-27, 2013.
- [8] S. Izadi and C. Y. Suen, "Online Writer-Independent Character Recognition Using a Novel Relational Context Representation," *Proc. Seventh International Conference on Machine Learning and Applications (ICMLA '08).*, pp. 867-870, Dec 11-13, 2008.
- [9] C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", *Data Mining and Knowledge Discovery*, Kluwer Academic Publishers, Vol . 2, pp. 121-167, Juni 1998.
- [10] M. Mori, *Charagter Recognition*, 1st ed, Sciyo, 2010.
- [11] A. N. Kurniawan, T. S. Widodo, dan I. Soesanti, "Penapisan Artifak Logam pada Citra CT-scan dengan Spatial Filter", *JNTETI*, Vol. 2, No. 4, Februari 2013.