Penggunaan *Decision Tree* Dengan *ID3 Algorithm* Untuk Mengenali Dokumen Beraksara Jawa

Bondan Sebastian, Gregorius Satia Budhi, Rudy Adipranata
Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto 121-131 Surabaya
Telp. (031) 2983455, Fax. (031) - 8417658
bondansebastian@gmail.com; greg@petra.ac.id; rudy@petra.ac.id

ABSTRAK

Skripsi ini bertujuan untuk mengenali huruf Jawa menggunakan komputer. Sebelum komputer dapat mengenali huruf Jawa, diperlukan proses segmentasi dan ekstraksi fitur dari gambar dokumen beraksara Jawa. Setelah fitur dari huruf Jawa yang telah tersegmentasi didapatkan, maka data fitur tersebut diolah dengan menggunakan *computer learning method* agar komputer dapat mengenali huruf-huruf tersebut.

Input berupa *file .CSV* yang berisi fitur –fitur dari gambar huruf Jawa yang sudah disegmentasi sebelumnya. *Output file text* yang berisi representasi *unicode* huruf Jawa dari *font* Hanacaraka dan sebuah *file .CSV* yang berisi hasil pengujian dan klasifikasi data. Aplikasi ini dibuat dengan bahasa pemrograman *C#* dengan *Microsoft Visual Studio* 2013 sebagai *IDE*-nya.

Adapun proses yang dilakukan adalah sebagai berikut: memproses *file .CSV* yang dihasilkan dari aplikasi ekstraksi fitur agar siap digunakan sebagai bahan pembelajaran atau *trainer* bagi *computer learning network*. Setelah *file trainer* siap, maka *network* akan di-*training*. Setelah proses *training* selesai, *network* dapat menerima input data lain, dan kemudian mengklasifikasikannya sesuai dengan apa yang telah dipelajari.

Computer learning network yang digunakan adalah *probabilistic neural network* dan *ID*, dimana setelah diuji diketahui bahwa untuk pengklasifikasian huruf Jawa, metode *PNN* mencapai hasil akurasi yang lebih tinggi daripada *ID3*

Kata Kunci: Huruf Jawa, *Decision Tree*, *ID3*

ABSTRACT

This thesis' goal is to let computer recognize Javanese letters. Before a computer is able to recognize Javanese letters, segmentation and feature extraction process is needed. After the features of Java letters that have been segmented obtained, then the features will be processed using computer learning method so that the computer can recognize the letters.

Input is in the form of .CSV file that contains features of the Javanese document image that had previously segmented. Output text is a text file that contains unicode representation letter from Java font and a .CSV file that contains the test and data classification results. This application is created with C# programming language and Microsoft Visual Studio 2013 IDE.

As for the process that is done is as follows: prepare the .CSV file resulting from the extraction of features to be used as learning materials or trainer for computer learning network. After the trainer file ready, then the network will be trained. After the training is complete, the network can receive other data input, and then classify it according to what has been learned.

Probabilistic neural network and ID3 used as computer learning methods, which after data testing, it is found that PNN give higher accuracy result than ID3.

Keywords: *Javanese letter, Decision Tree , ID3*

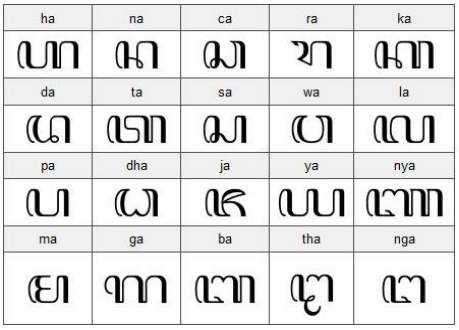
1. LATAR BELAKANG

Dari penelitian sebelumnya (Aplikasi Ekstraksi Fitur Citra Huruf Jawa Berdasarkan Morfologinya, Meiliana Indrawijaya, 2014) telah dihasilkan aplikasi yang dapat mengekstrak fitur aksara jawa yang berdasarkan ciri bentuknya (morfologinya). Setelah fitur aksara jawa telah diekstrak ke dalam bentuk data, maka dibutuhkan jaringan syaraf tiruan untuk dilatih agar komputer dapat mengenali berbagai aksara jawa lewat fiturnya. Pada penelitian ini, peneliti akan mengembangkan aplikasi yang dilengkapi jaringan syaraf tiruan untuk dapat mengenali huruf jawa berdasarkan fitur yang telah didapat. Sebelumnya pernah ada penelitian serupa mengenai klasifikasi huruf Jawa, yang berjudul "Konversi Huruf Hanacaraka Ke Huruf Latin Menggunakan Metode *Modified Direction Feature* (*MDF*) Dan *K-Nearest Neighbour* (*KNN*)" [3], dimana sebelum dikonversi menjadi huruf latin, huruf Jawa terlebih dahulu diklasifikasikan dengan *MDF* dan *KNN*. Hasil klasifikasi dengan kedua metode tersebut disimpulkan tidak akurat. Kali ini, metode jaringan syaraf tiruan yang digunakan dalam klasifikasi adalah *decision tree* dengan *ID3* sebagai algoritma penyusun *decision tree*.

2. LANDASAN TEORI

2.1 Huruf Jawa

Aksara Jawa berbeda dengan huruf Latin yang biasa digunakan. aksara Jawa terdiri dari aksara Carakan [1], aksara Pasangan, aksara Swara, aksara Rekan, aksara Murda, aksara Wilangan dan Pelengkap/Sandhangan. [2]. Gambar contoh aksara Jawa dapat dilihat pada Gambar 1.



Gambar 1. Contoh Aksara Jawa

2.2 Segmentasi Huruf Jawa

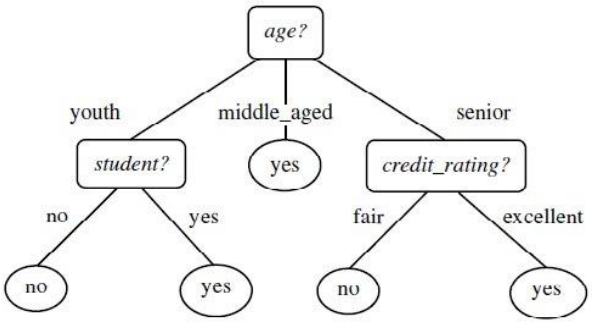
Segmentasi huruf Jawa adalah proses dimana komputer mendeteksi setiap huruf Jawa dari suatu dokumen. Dari *input* yang berupa gambar dokumen beraksara Jawa, akan dihasilkan *output* berupa file *CSV* (*Comma Separated Value*). Data yang disimpan di dalam file *CSV* adalah koordinat x dan y suatu huruf terhadap titik pojok kanan atas dokumen, serta panjang (*width*) dan tinggi (*height*) huruf tersebut dalam satuan pixel. [7]

2.3 Ekstraksi Fitur Huruf Jawa

Setiap huruf Jawa memiliki jumlah lengkungan dan garis lurus yang tidak sama. Setiap lengkungan (*curve*), garis lurus (*line*), dan bulatan penuh (*loop*) yang dimiliki oleh masing-masing huruf disebut dengan fitur. Perbedaan fitur inilah yang akan digunakan untuk membedakan antara huruf Jawa yang satu dengan yang lain. [6]

2.4 *Decision Tree*

Decision Tree Induction adalah pembelajaran *decision tree* dari *class-labeled training tuples*. *Decision tree* berbentuk seperti *flowchart* di mana setiap *internal node* menggambarkan sebuah tes pada sebuah atribut, setiap *branch* menggambarkan hasil dari tes, dan setiap *leaf node* menunjukkan *class label*. *Node* paling atas adalah *root node* [4]. Contoh dari *decision tree* dapat dilihat pada Gambar 2. Gambar ini menunjukkan prediksi apakah seorang *customer* di *AllElectronics* akan membeli komputer atau tidak. *Internal node* digambarkan dengan kotak, dan *leaf node* digambarkan dengan oval.



Gambar 2. Contoh *Decision Tree*

Selama membuat *tree*, perlu memilih atribut yang paling baik dalam membedakan kelas-kelas. Untuk itu digunakan algoritma. algoritma *decision* tree yang digunakan adalah *ID3*.

2.5 *ID3* (*Iterative Dichotomiser*)

ID3 (*Iterative Dichotomiser*) merupakan sebuah algoritma pohon keputusan yang dibuat oleh J. Ross Quinlan [8]. ID3 menggunakan algoritma *basic tree induction* yang memberi atribut ke node pada *tree* berdasar berapa banyak informasi bertambah dari node tersebut. Metode ID3 memperbolehkan sebuah atribut untuk memiliki dua atau lebih nilai pada sebuah node atau titik split [1]. Atribut yang dipilih untuk setiap node adalah atribut yang memaksimalkan *information gain*. ID3 dapat mengklasifikan data yang besar dalam waktu yang relatif cepat, tergantung seberapa besar *data set* yang digunakan [5].

2.6 *Information Gain*

ID3 menggunakan *information gain* untuk pemilihan atributnya. Atribut yang memiliki *information gain* tertinggi dipilih menjadi atribut *splitting* untuk sebuah *node* [8]. Informasi yang diperkirakan dibutuhkan untuk mengklasifikasi sebuah *tuple* dalam D diberikan :

$$Info(D) = -\sum_{i=1}^{m} p_1 \log 2\,(p_1) \qquad (1)$$

di mana p_i adalah kemungkinan sebuah *arbitrary tuple* dalam D termasuk kelas C_i dan diestimasi oleh $|Ci_{,D}\ |/|D|$. Fungsi log dengan basis 2 digunakan karena informasi dihitung dalam bit. *Info(D)* hanya jumlah rata-rata informasi yang digunakan untuk mengidentifikasi *class label* dari sebuah *tuple* dalam D. *Info(D)* dikenal sebagai *entropy* dari D. Jumlah informasi yang masih dibutuhkan untuk mencapai partisi yang tepat diukur dengan persamaan [8].

$$info_A(D) = a_0 + \sum_{j-l}^{v} \frac{|D_j|}{|D|} Info(D_j) \qquad (2)$$

Term $\frac{|Dj|}{|D|}$ berfungsi sebagai bobot dari partisi ke j. *InfoA(D)* adalah perkiraan informasi yang dibutuhkan untuk mengklasifikasikan sebuah *tuple* dari D berdasarkan partisi oleh A. Semakin kecil informasi yang diperkirakan dibutuhkan, semakin besar tingkat kemurnian dari partisi. *Information gain* didefinisikan sebagai perbedaan antara kebutuhan informasi asal dengan kebutuhan yang baru [8]. Yaitu :

$$Gain(A) = Info(D) - Info_A(D) \qquad (3)$$

Dengan kata lain, Gain(A) memberi tahu seberapa banyak akan bertambah dengan *branching* di A. Atribut A dengan *information gain* terbesar, (Gain(A)) akan dipilih sebagai atribut *splitting* di *node* yang bersangkutan.

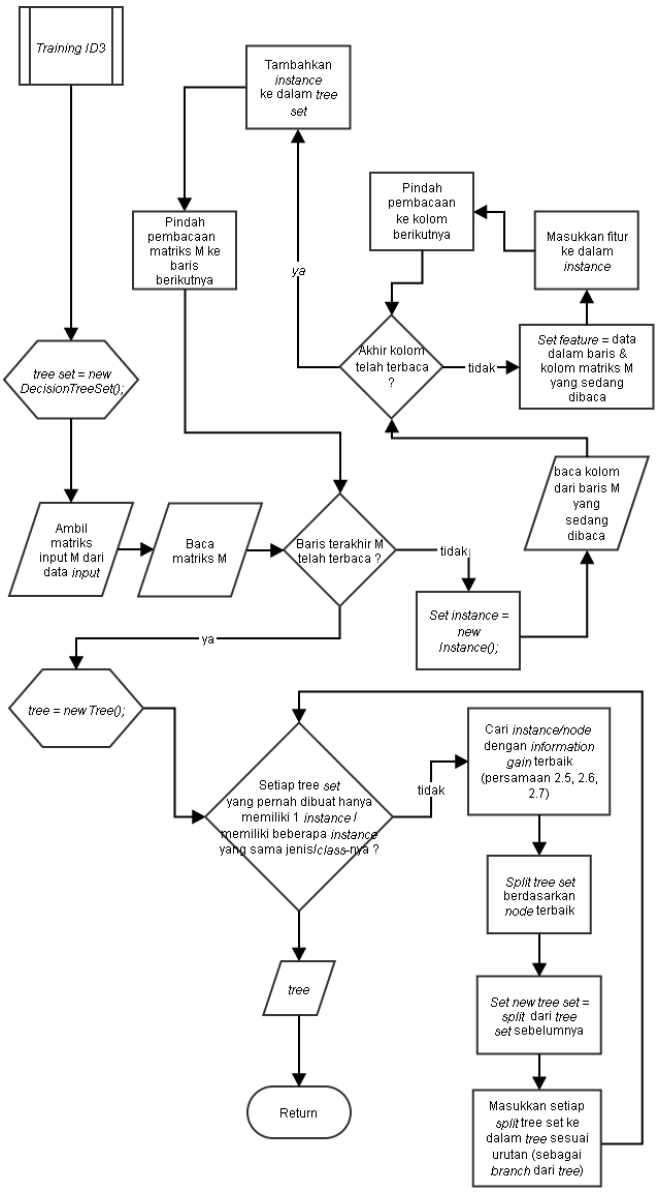
3. DESAIN SISTEM

3.1 Garis Besar Sistem Kerja Perangkat Lunak

Software pengenalan huruf Jawa ini dapat dibagi menjadi dua proses utama, yaitu proses *training* (dalam konteks ini penyusunan *decision tree*) dan klasifikasi.

3.2 Garis Besar Proses *Training*

Untuk melakukan proses *training* terdapat beberapa tahapan yang harus dilakukan. Adapun rancangan sistem kerja perangkat lunak untuk modul *training* secara garis besar ditunjukan oleh Gambar 3.



Gambar 3. Diagram alur proses *training*

4. IMPLEMENTASI SISTEM

Dalam pembuatan skripsi ini, perwujudan dari *decision tree* adalah sebuah *class* yang diberi nama *Tree*.

Dalam sebagian operasinya, *class* ini bergantung pada beberapa *class* lain dalam *namespace ClassifierNetwork.ID3*. Karena itu *class* beberapa *class* lain dalam *namespace ClassifierNetwork.ID3* juga akan disinggung dalam bagian ini. Terdapat 2 proses penting yang berkaitan dengan *class Tree*, yaitu proses pembentukan *tree* dan proses klasifikasi data berdasarkan *rule* yang telah dibentuk oleh *tree*. Kedua proses ini direpresentasikan oleh fungsi *BuildTree* dan *Classify*.

Fungsi *BuildTree* tidak berada dalam *class Tree*, namun berada dalam *class DecisionTreeSet*, karena sebelum dibentuk menjadi suatu *rule tree*, semua fitur huruf Jawa terlebih dahulu dikumpulkan dan ditampung dalam *class DecisionTreeSet*. Fungsi *BuildTree* merukapan fungsi rekursif, dimana fungsi ini akan berhenti ketika dalam suatu *DecisionTreeSet* hanya terdapat satu *instance* saja, atau beberapa *instance* dengan kategori yang sama.

Fungsi *BuildTree* bekerja dengan cara memilih fitur terbaik untuk dijadikan titik *split* dalam *tree*, kemudian dari titik ini akan terjadi percabangan *branch* dalam *tree* berdasarkan fitur-fitur huruf Jawa. Untuk memilih fitur terbaik, fungsi *BuildTree* akan memanggil fungsi *SelectBestAxis* yang merupakan salah satu fungsi dari *class Decider*. Sekilas mengenai *class Decider*, *class* ini merupakan *static class* yang berisi berbagai fungsi untuk mendukung pembangunan *decision tree*.

Setelah diketahui nilai *entropy* dari suatu *set* data, maka dilakukan penghitungan nilai *entropy* dari setiap *cabang* yang mungkin dibentuk oleh *set* data sebelumnya. Untuk menghitung nilai *entropy* dari suatu cabang, digunakan fungsi *EntropyForSplitBranches*.

Setelah diketahui nilai *entropy* dari keseluruhan suatu *set* data (sebut saja *base entropy)* dan nilai *entropy* dari setiap cabang yang mungkin dibentuk oleh *set* tersebut, maka setiap *base entropy* akan dikurangkan dengan setiap nilai *entropy* dari cabang yang mungkin terbentuk, menghasilkan nilai yang disebut *info gain*. *Axis* atau fitur yang menghasilkan cabang dengan *info gain* paling besar akan dipilih sebagai titik *split* terbaik.

Setelah proses pemilihan *axis* terbaik untuk dijadikan titik *split*, maka proses *splitting* yang sebenarnya akan dilakukan. Proses *splitting* dilakukan oleh fungsi *SplitByAxis*. Secara garis besar, fungsi ini akan membentuk suatu *Tree* dengan *axis* terbaik sebagai *root node*-nya. Dalam fungsi ini, fungsi *BuildTree* kembali dipanggil untuk membentuk cabang baru dalam bentuk *tree.*

Dalam fungsi *SplitByAxis*, terdapat fungsi *UniqueFeatures*. Seperti namanya, fungsi ini berguna untuk mendapatkan fitur-fitur unik dari suatu *instance*. Fitur unik yang dimaksudkan adalah fitur-fitur yang berbeda satu sama lain (tidak kembar).

Selain fungsi *UniqueFeatures,* fungsi *SplitByAxis* juga memanggil fungsi lain, yaitu fungsi *Split*. Fungsi-fungsi yang telah disebutkan sebelumnya adalah fungsi-fungsi yang berhubungan langsung dengan proses *training* atau pembentukan *decision* tree.

5. PENGUJIAN SISTEM

5.1 Pengujian Akurasi Metode *ID3* terhadap Berbagai Perubahan Parameter Data yang Diuji

Pengujian ini akan dimulai dengan pengujian terhadap data yang sudah di-*training*-kan terlebih dahulu. Hasil dari pengujian dengan data yang sudah di-*training*-kan dapat dilihat pada Tabel 1.

Tabel 1. Hasil pengujian *ID3* dengan data yang sudah di-*training*-kan

No.	Nama gambar	Jumlah data	Akurasi (dalam %)
1	File4.jpg	136	100
2	IMG_0011.jpg	55	100
3	P5080088.jpg	177	100
4	P5080214.jpg	212	100
5	P5080220.jpg	283	100
6	P5080223.jpg	198	100

Tabel 1. Hasil pengujian *ID3* dengan data yang sudah di-*training*-kan [lanjutan]

No.	Nama gambar	Jumlah data	Akurasi (dalam %)
7	P5080226.jpg	196	100
8	P5100441.jpg	25	100
9	P5100442.jpg	130	100
10	P5100443.jpg	156	100
Rata - rata			**100**

Sebelum ke tahap pengujian berikutnya, akan diuji klasifikasi *ID3* dengan data yang dipisahkan sesuai dengan jenisnya. Pertama akan diuji data carakan yang sudah di-*training*-kan sebelumnya. Pengujian dapat dilihat pada Tabel 2.

Tabel 2. Hasil pengujian *ID3* dengan data yang sudah di-*training*-kan (carakan)

No.	Nama gambar	Jumlah data	Akurasi (dalam %)
1	File4.jpg	58	100
2	IMG_0011.jpg	25	100
3	P5080088.jpg	87	100
4	P5080214.jpg	94	100
5	P5080220.jpg	132	100
6	P5080223.jpg	122	100
7	P5080226.jpg	90	100
8	P5100441.jpg	10	100
9	P5100442.jpg	60	100
10	P5100443.jpg	86	100
Rata - rata			**100**

Setelah pengujian dengan aksara carakan yang telah di-*training*-kan sebelumnya, kemudian diuji data aksara pasangan yang telah di-*training*-kan sebelumnya. Hasil pengujian dapat dilihat pada Tabel 3.

Tabel 3. Hasil pengujian *ID3* dengan data yang sudah di-*training*-kan (pasangan)

No.	Nama gambar	Jumlah data	Akurasi (dalam %)
1	File4.jpg	45	100
2	IMG_0011.jpg	17	100
3	P5080088.jpg	53	100
4	P5080214.jpg	64	100
5	P5080220.jpg	85	100
6	P5080223.jpg	59	100
7	P5080226.jpg	59	100
8	P5100441.jpg	8	100
9	P5100442.jpg	39	100
10	P5100443.jpg	47	100
Rata - rata			**100**

Setelah pengujian dengan data aksara pasangan, sekarang *ID3* diuji dengan data aksara sandhangan yang sudah di-*training*. Hasil pengujian dapat dilihat pada Tabel 4.

Setelah pengujian dengan data yang sudah di-*training*-kan, dilakukan pengujian dengan data yang belum pernah di-*training*-kan. Data *training* diambil dari 10 gambar dalam tabel 5.1, akan tetapi dibatasi setiap huruf yang sama dalam gambar hanya boleh diambil maksimal 2 sebagai data *training*. Sisanya akan dipakai sebagai data klasifikasi. Hasil pengujian dapat dilihat pada Tabel 5.

Tabel 4. Hasil pengujian *ID3* dengan data yang sudah di-*training*-kan (sandhangan)

No.	Nama gambar	Jumlah data	Akurasi (dalam %)
1	File4.jpg	45	100
2	IMG_0011.jpg	17	100
3	P5080088.jpg	53	100
4	P5080214.jpg	64	100
5	P5080220.jpg	85	100
6	P5080223.jpg	59	100
7	P5080226.jpg	59	100
8	P5100441.jpg	8	100
9	P5100442.jpg	39	100
10	P5100443.jpg	47	100
Rata - rata			**100**

Tabel 5. Hasil pengujian *ID3* dengan data yang belum pernah di-*training*-kan

No.	Nama gambar	Jumlah data *training*	Jumlah data klasifikasi	Akurasi (dalam %)
1	File4.jpg	57	79	7,59
2	IMG_0011.jpg	28	27	22,22
3	P5080088.jpg	51	126	7,94
4	P5080214.jpg	59	153	15,69
5	P5080220.jpg	61	222	12,61
6	P5080223.jpg	56	142	8,45
7	P5080226.jpg	60	136	12,50
8	P5100441.jpg	15	10	20
9	P5100442.jpg	43	87	27,59
10	P5100443.jpg	46	110	6,36
Rata - rata				**14,1**

Sekarang akan diuji tingkat akurasi *ID3* dengan data yang telah dipisah-pisah sesuai dengan jenisnya. Pertama – tama akan diuji data aksara carakan yang belum di-*training*-kan. Hasil pengujian dapat dilihat pada Tabel 6.

Tabel 6. Hasil pengujian *ID3* dengan data yang belum pernah di-*training*-kan (carakan)

No.	Nama gambar	Jumlah data *training*	Jumlah data klasifikasi	Akurasi (dalam %)
1	File4.jpg	25	33	13,14
2	IMG_0011.jpg	13	12	14,89
3	P5080088.jpg	20	67	15,32
4	P5080214.jpg	23	71	14,67
5	P5080220.jpg	31	101	14,89
6	P5080223.jpg	27	95	13,87
7	P5080226.jpg	33	57	14,11
8	P5100441.jpg	5	5	13,87
9	P5100442.jpg	23	37	14,55
10	P5100443.jpg	23	63	14,56
Rata - rata				**14,39**

Setelah pengujian dengan aksara carakan yang belum di-*training*-kan sebelumnya, kemudian diuji data aksara pasangan yang belum di-*training*-kan sebelumnya. Hasil pengujian dapat dilihat pada Tabel 7.

Tabel 7. Hasil pengujian *ID3* dengan data yang belum pernah di-*training*-kan (pasangan)

No.	Nama gambar	Jumlah data *training*	Jumlah data klasifikasi	Akurasi (dalam %)
1	File4.jpg	12	21	12,52
2	IMG_0011.jpg	6	7	13,22
3	P5080088.jpg	14	23	14,60
4	P5080214.jpg	20	34	13,98
5	P5080220.jpg	11	55	14,19
6	P5080223.jpg	5	12	14,19
7	P5080226.jpg	13	34	13,45
8	P5100441.jpg	5	2	13,22
9	P5100442.jpg	7	24	13,87
10	P5100443.jpg	10	13	13,88
Rata - rata				**13,73**

Setelah pengujian dengan aksara pasangan yang belum di-*training*-kan, sekarang akan diujikan data sandhangan yang belum pernah di-*training*-kan. Hasil pengujian dapat dilihat pada Tabel 8.

Tabel 8. Hasil pengujian *PNN* dengan data yang belum pernah di-*training*-kan (sandhangan)

No.	Nama gambar	Jumlah data *training*	Jumlah data klasifikasi	Akurasi (dalam %)
1	File4.jpg	20	25	12,80
2	IMG_0011.jpg	9	8	14,50
3	P5080088.jpg	17	36	14,92
4	P5080214.jpg	16	48	14,29
5	P5080220.jpg	19	66	14,50
6	P5080223.jpg	24	35	13,51
7	P5080226.jpg	14	45	13,74
8	P5100441.jpg	5	3	13,51
9	P5100442.jpg	13	26	14,17
10	P5100443.jpg	13	34	14,18
Rata - rata				**14,03**

Berdasarkan hasil pengujian yang telah dilakukan, untuk data yang telah dipisahkan sesuai dengan jenisnya, dengan data yang telah di-*training*-kan sebelumnya *ID3* dapat mencapai akurasi 100% untuk setiap jenis huruf, sedangkan untuk data yang belum di-*training*-kan hasil akurasi tertinggi ditemukan pada aksara carakan, dan hasil akurasi terendah ditemukan pada aksara pasangan. Untuk data yang tidak dipisahkan, dapat dilihat bahwa tingkat akurasi rata-rata *ID3* dengan data yang sudah di-*training*-kan mencapai 100%, sedangkan tingkat akurasi rata-ratanya untuk data yang belum pernah di-*training*-kan mencapai 14,1%.

6. KESIMPULAN

Berdasarkan hasil pengujian, dapat ditarik beberapa kesimpulan sebagai berikut :

- Untuk data yang sudah pernah di-*training*-kan, metode *ID3* dapat mencapai akurasi sampai 100% terhadap semua jenis data.
- Untuk data yang belum pernah di-*training*-kan, metode *ID3* dapat mencapai akurasi sampai 14,39% terhadap aksara carakan, 13,73% terhadap aksara pasangan, 14,03% untuk aksara sandhangan, dan 14,1% untuk keseluruhan data.
- Metode *ID3* kurang cocok untuk dipakai dalam panegklasifikasian huruf Jawa.
- Metode akstraksi fitur dari penelitian sebelumnya tidak cocok untuk diaplikasikan pada jaringan syaraf tiruan, karena menghasilkan data yang terlalu *uniform* bagi jaringan syaraf tiruan.

7. REFERENSI

[1] Berry, M.W. & Browney, M. 2006. *Lectures note in data mining.* USA: World Scientific Publishing Co. Pte. Ltd.

[2] Digdadinaya. 2014. *Belajar Bersama Aksara Jawa*. URI = http://www.kaskus.co.id/thread/53e438f5925233464e8b45b2/share-belajar-bersama-aksara-jawa--hanacaraka/

[3] Feriyanto. 2014. *Konversi Huruf Hanacaraka ke Huruf Latin Menggunakan Metode Modified Direction Feature (MDF) dan K-Nearest Neighbour (KNN)*. Jurnal, Universitas Teknologi Bandung.

[4] Han, J., & Kamber, M. 2006. *Data mining: Concepts and techniques (2nd ed.).* San Fransisco, CA: Morgan Kaufmann.

[5] Han, J., Kamber, M., Pei, J. 2012. *Data mining: Concepts and techniques (3nd ed.).* San Fransisco, CA: Morgan Kaufmann.

[6] Indrawijaya, M. 2015. *Aplikasi Ekstraksi Fitur Citra Huruf Jawa Berdasarkan Morfologinya*. Skripsi. 01021378/INF/2015, Universitas Kristen Petra.

[7] Mardianto, S. 2015. *Aplikasi Segmentasi Huruf Jawa*. Skripsi. Universitas Kristen Petra.

[8] Putra, C.A. 2012. *Perancangan dan Pembuatan Aplikasi Klasifikasi Citra Observasi Bintik Matahari Menggunakan Metode ID3 dan C4.5*. Skripsi. 01021111/INF/2012, Universitas Kristen Petra.