Penerapan *Naive Bayes* untuk Mengurangi Data *Noise* pada Klasifikasi Multi Kelas dengan *Decision Tree*

Al Riza Khadafy
Program Studi Ilmu Komputer, STMIK Nusa Mandiri Jakarta
rizacalm@gmail.com

Romi Satria Wahono
Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
romi@romisatriawahono.net

Abstrak: Selama beberapa dekade terakhir, cukup banyak algoritma *data mining* yang telah diusulkan oleh peneliti kecerdasan komputasi untuk memecahkan masalah klasifikasi di dunia nyata. Di antara metode-metode *data mining* lainnya, *Decision Tree* (DT) memiliki berbagai keunggulan diantaranya sederhana untuk dipahami, mudah untuk diterapkan, membutuhkan sedikit pengetahuan, mampu menangani data numerik dan kategorikal, tangguh, dan dapat menangani *dataset* yang besar. Banyak *dataset* berukuran besar dan memiliki banyak kelas atau multi kelas yang ada di dunia memiliki *noise* atau mengandung *error*. Algoritma pengklasifikasi DT memiliki keunggulan dalam menyelesaikan masalah klasifikasi, namun data *noise* yang terdapat pada *dataset* berukuran besar dan memiliki banyak kelas atau multi kelas dapat mengurangi akurasi pada klasifikasinya. Masalah data *noise* pada *dataset* tersebut akan diselesaikan dengan menerapkan pengklasifikasi *Naive Bayes* (NB) untuk menemukan *instance* yang mengandung *noise* dan menghapusnya sebelum diproses oleh pengklasifikasi DT. Pengujian metode yang diusulkan dilakukan dengan delapan *dataset* uji dari UCI (*University of California, Irvine*) *machine learning repository* dan dibandingkan dengan algoritma pengklasifikasi DT. Hasil akurasi yang didapat menunjukkan bahwa algoritma yang diusulkan DT+NB lebih unggul dari algoritma DT, dengan nilai akurasi untuk masing-masing *dataset* uji seperti *Breast Cancer* 96.59% (meningkat 21,06%), *Diabetes* 92,32% (meningkat 18,49%), *Glass* 87,50% (meningkat 20,68%), *Iris* 97,22% (meningkat 1,22%), *Soybean* 95,28% (meningkat 3,77%), *Vote* 98,98% (meningkat 2,66%), *Image Segmentation* 99,10% (meningkat 3,36%), dan *Tic-tac-toe* 93,85% (meningkat 9,30%). Dengan demikian dapat disimpulkan bahwa penerapan NB terbukti dapat menangani data *noise* pada *dataset* berukuran besar dan memiliki banyak kelas atau multi kelas sehingga akurasi pada algoritma klasifikasi DT meningkat.

Keywords: data *noise*, pengklasifikasi *Naive Bayes*, pengklasifikasi *Decision Tree*

1 PENDAHULUAN

Selama beberapa dekade terakhir, cukup banyak algoritma *data mining* yang telah diusulkan oleh peneliti kecerdasan komputasi untuk memecahkan masalah klasifikasi di dunia nyata (Farid et al., 2013; Liao, Chu, & Hsiao, 2012; Ngai, Xiu, & Chau, 2009). Secara umum, klasifikasi adalah fungsi *data mining* yang menggambarkan dan membedakan kelas data atau konsep. Tujuan dari klasifikasi adalah untuk secara akurat memprediksi label kelas dari *instance* yang nilai atributnya diketahui, tapi nilai kelasnya tidak diketahui. Beberapa algoritma *data mining* yang sering digunakan untuk klasifikasi diantaranya adalah *Decision Tree* dan *Naive Bayes*.

Decision Tree (DT) atau pohon keputusan adalah algoritma klasifikasi yang banyak digunakan dalam *data mining* seperti ID3 (Quinlan, 1986), ID4 (Utgoff, 1989), ID5 (Utgoff, 1989), C4.5 (Quinlan, 1993), C5.0 (Bujlow, Riaz, & Pedersen, 2012), dan CART (Breiman, Friedman, Olshen, & Stone, 1984). Tujuan dari DT adalah untuk membuat model yang dapat memprediksi nilai dari sebuah kelas target pada *test instance* yang tidak terlihat berdasarkan beberapa fitur masukan (Loh & Shih, 1997; Safavian & Landgrebe, 1991; Turney, 1995). Di antara metode-metode *data mining* lainnya, DT memiliki berbagai keunggulan diantaranya sederhana untuk dipahami, mudah untuk diterapkan, membutuhkan sedikit pengetahuan, mampu menangani data numerik dan kategorikal, tangguh, dan dapat menangani *dataset* yang besar (Han, Kamber, & Pei, 2012).

Berbagai metode terkait algoritma pengklasifikasi DT telah dikembangkan pada beberapa penelitian, diantaranya adalah *Decision Tree Using Fast Splitting Attribute Selection* (DTFS) (Franco-Arcega, Carrasco-Ochoa, Sanchez-Diaz, & Martinez-Trinidad, 2011), *Classification by Clustering* (CbC) (Aviad & Roy, 2011), C4.5 dengan pendekatan *One-Against-All* untuk meningkatkan akurasi klasifikasi pada masalah klasifikasi multi kelas (Polat & Gunes, 2009), penanganan eksepsi pada DT (Balamurugan & Rajaram, 2009), *Associative Classification Tree* (ACT) (Chen & Hung, 2009), *Fuzzy Decision Tree Gini Index Based* (G-FDT) (Chandra & Paul Varghese, 2009), dan *Co-Evolving Decision Tree* (Aitkenhead, 2008).

Performa algoritma *data mining* dalam banyak kasus tergantung pada kualitas *dataset*, karena data *training* berkualitas rendah dapat menyebabkan klasifikasi yang lemah (Han et al., 2012). Dengan demikian, dibutuhkan teknik *data preprocessing* untuk mempersiapkan data yang akan diproses. Hal ini dapat meningkatkan kualitas data, sehingga membantu untuk meningkatkan akurasi dan efisiensi proses *data mining*. Beberapa teknik *data preprocessing* diantaranya adalah *data cleaning*: menghapus data yang mengandung *error*, *data integration*: menggabungkan data dari berbagai sumber, *data transformation*: normalisasi data, dan *data reduction*: mengurangi ukuran data dengan menggabungkan dan menghilangkan fitur yang berlebihan.

Naive Bayes (NB) adalah algoritma klasifikasi probabilitas sederhana yang berdasarkan pada teorema Bayes, asumsi bebas yang kuat (*naive*), dan model fitur independen (Farid, Rahman, & Rahman, 2011; Farid & Rahman, 2010; Lee & Isa, 2010). NB juga merupakan algoritma klasifikasi yang utama pada *data mining* dan banyak diterapkan dalam masalah klasifikasi di dunia nyata karena memiliki performa klasifikasi yang

tinggi. Mirip dengan DT, algoritma pengklasifikasi NB juga memiliki beberapa keunggulan seperti mudah digunakan, hanya membutuhkan satu kali *scan* data *training*, penanganan nilai atribut yang hilang, dan data kontinu (Han et al., 2012).

Banyak *dataset* berukuran besar dan memiliki banyak kelas atau multi kelas yang ada di dunia memiliki *noise* atau mengandung *error*, hal ini dapat menyebabkan berkurangnya akurasi pada klasifikasi DT (Han et al., 2012; Polat & Gunes, 2009; Quinlan, 1986). *Instance* yang mengandung *error* pada *dataset* menyebabkan salah klasifikasi saat diproses oleh algoritma pengklasifikasi NB. Dengan demikian, algoritma pengklasifikasi NB dapat digunakan untuk menemukan *instance* yang bermasalah pada *dataset*.

Pada penelitian ini algoritma pengklasifikasi NB akan digunakan untuk menemukan *instance* yang bermasalah pada data *training* dan menghapusnya sebelum algoritma DT membuat pohon keputusan agar akurasi klasifikasinya meningkat.

2 PENELITIAN TERKAIT

Polat dan Gunes melakukan penelitian pada tahun 2009, yaitu mereka menggabungkan algoritma pengklasifikasi C4.5 dengan pendekatan *One-Against-All* untuk memecahkan masalah klasifikasi multi kelas. Pada penelitian tersebut digunakan *dataset* dari UCI *machine learning repository*, diantaranya *dataset Dermatology*, *Image Segmentation*, dan *Lymphography*. Pertama algoritma C4.5 dijalankan pada setiap *dataset* menggunakan 10-*fold cross validation* dan mendapatkan hasil akurasi 84,48%, 88,79%, dan 80,11% pada masing-masing *dataset*. Kemudian algoritma usulan dijalankan pada setiap *dataset* menggunakan 10-*fold cross validation* dan mendapatkan hasil akurasi yang lebih tinggi yaitu 96,71%, 95,18%, dan 87,95% pada masing-masing *dataset*.

Penelitian yang dilakukan Aitkenhead pada tahun 2008, yaitu dengan mengembangkan pendekatan evolusioner pada algoritma *Decision Tree* untuk mengatasi masalah data *noise* dan masalah kombinasi data kuantitatif dan kualitatif pada *dataset* yang dapat menyulitkan proses kategorisasi kelas. Pada penelitian tersebut digunakan *dataset Glass Chemistry* dan *Car Costing*. Pengujian dilakukan dengan menjalankan algoritma usulan pada setiap *dataset*, kemudian dibandingkan dengan algoritma C4.5 dan didapatkan hasil akurasi yang lebih tinggi yaitu 0,824 dan 0,892 pada masing-masing *dataset*.

Penelitian yang dilakukan Balamurugan dan Rajaram pada tahun 2009, yaitu mereka melakukan perbaikan pada algoritma *Decision Tree* dengan menambahkan prosedur penghitungan *Maximum Influence Factor* (MIF) untuk mengatasi masalah kegagalan dalam pemilihan atribut yang akan di-*split* yang dapat menyebabkan label kelas dipilih secara acak. *Dataset* yang digunakan dalam penelitian tersebut diantaranya *Blood Transfusion*, *Teaching Assistant Evaluation*, *SPECT Heart*, *Haberman's Survival*, *Contraceptive Method Choice*, *Hayes Roth*, *Concrete*, *Forest-fires*, *Solarflare 1*, dan *Solarflare 2*. Pengujian dilakukan dengan menjalankan algoritma usulan pada setiap *dataset* kemudian dilakukan perbandingan dengan algoritma lain seperti C4.5, NB, K-NN. Pada penelitian tersebut didapatkan nilai akurasi lebih tinggi yaitu 85,16 %, 77,78 %, 71,70 %, 78,79 %, 77,50 %, 76,74 %, 76,74 %, 75,68 %, 77,09% pada masing-masing *dataset*.

Penelitian yang dilakukan Chandra dan Paul Varghese pada tahun 2009, yaitu mereka melakukan perbaikan terhadap algoritma *Decision Tree* untuk mengatasi masalah pemilihan *splitting attribute* yang dapat menyebabkan *misclassification*. Perbaikan yang dilakukan adalah dengan menggunakan teknik *Fuzzy Decision Tree Algorithm Gini Index Based* (G-FDT). *Dataset* yang digunakan dalam penelitian tersebut diantaranya *Haberman*, *Iris*, *Balanced Scale*, *Liver*, *Diabetes*, *Wincosin BC*, *Echocardiogram*, *Wine*, *Ionosphere*, *Glass*, *Vehicle Silhouette*, *Heart Stat Log*, *Smoking*, *Contraceptive Method Choice*. Pengujian dilakukan dengan menjalankan algoritma usulan pada setiap *dataset* kemudian dilakukan perbandingan dengan algoritma *Supervised Learning In Quest* (SLIQ). Pada penelitian tersebut didapatkan hasil akurasi dan kecepatan algoritma yang lebih tinggi.

3 METODE USULAN

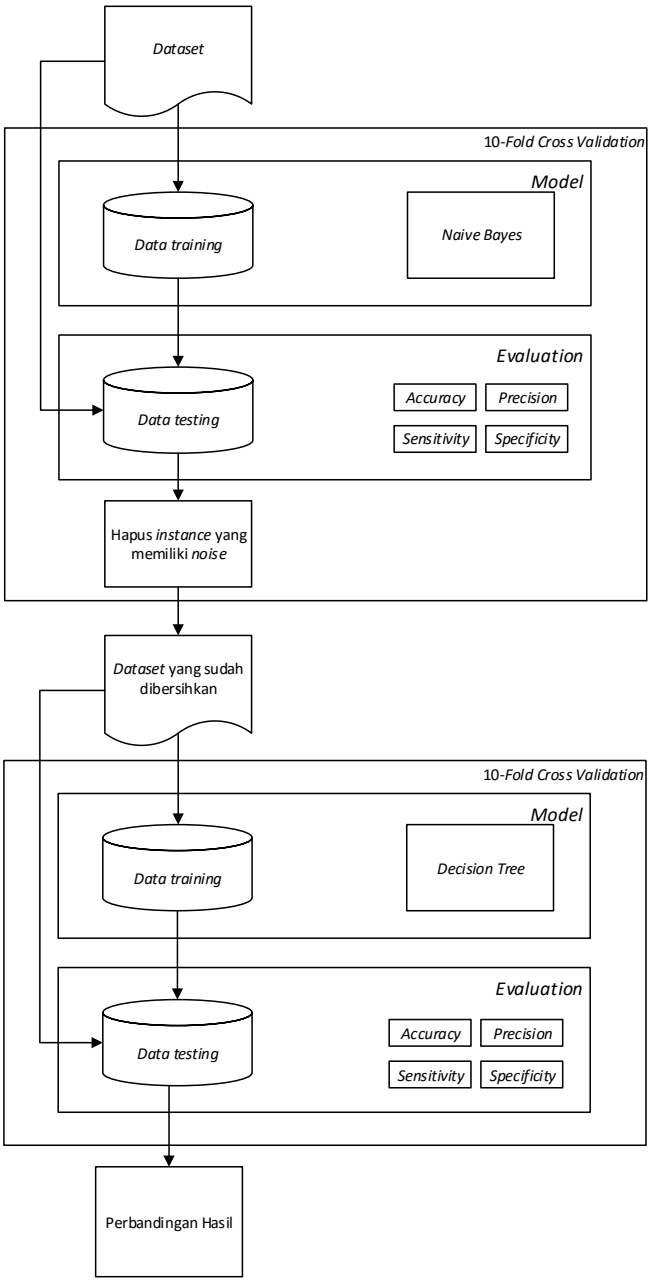
Untuk menangani masalah data *noise* pada klasifikasi *Decision Tree* (DT), diusulkan metode dengan pendekatan klasifikasi *Naive Bayes* (NB) untuk menemukan *instance* yang bermasalah atau mengandung *noise* kemudian menghapus *instance* tersebut. *Pseudocode* algoritma usulan ditunjukkan pada Gambar 1.

```
    Input: D = {x1, x2, ..., xn} // dataset training
    Output: T, Decision tree. // model decision tree
    Metode:
1   for each class, Ci ∈ D, do
2       Find the prior probabilities, P(Ci).
3   end for
4   for each attribute value, Aij ∈ D, do
5       Find the class conditional probabilities, P(Aij|Ci).
6   end for
7   for each training instance, xi ∈ D, do
8       Find the posterior probabilities, P(Ci|xi)
9       if xi misclassified, do
10          Remove xi from D; // hapus instance yang salah klasifikasi
11      end if
12  end for
13  T = ∅;
14  Determine best splitting attribute;
15  T = Create the root node and label it with the splitting attribute;
16  T = Add arc to the root node for each split predicate and label;
17  for each arc do
18      D = Dataset created by applying splitting predicate to D;
19      if stopping point reached for this path,
20          T' = Create a leaf node and label it with an appropriate class;
21      else
22          T' = DTBuild(D);
23      end if
24      T = Add T' to arc;
25  end for
```

Gambar 1. *Pseudocode* Algoritma DT + NB

Perancangan metode yang diusulkan yaitu dengan menerapkan algoritma pengklasifikasi NB untuk mengurangi *noise* pada klasifikasi multi kelas dengan DT. Dimulai dengan membagi *dataset* menjadi data *training* dan data *testing* dengan menggunakan metode 10-*fold cross validation,* kemudian menerapkan algoritma pengklasifikasi NB untuk menemukan dan kemudian menghapus *instance* yang memiliki *noise*. Kemudian *dataset* yang sudah dibersihkan dari *instance* yang memiliki *noise* tersebut diproses menggunakan algoritma DT untuk menghasilkan pohon keputusan. Selanjutnya hasil evaluasi model diukur nilai *accuracy*, *precision*, *sensitivity*, dan *specificity*. Gambar 2 menampilkan metode yang diusulkan.

Metode yang diusulkan diawali dengan membagi *dataset* menjadi data *training* dan data *testing* dengan menggunakan 10-*fold cross validation*, yaitu dengan membagi data 90% untuk proses *training* dan 10% untuk proses *testing*. Data *training* diproses dengan menggunakan algoritma pengklasifikasi NB untuk menghasilkan model klasifikasi. Kemudian dengan model klasifikasi tersebut dilakukan *testing*. Selanjutnya *instance* yang ditemukan salah klasifikasi atau *misclassified* dihapus dari *dataset*.

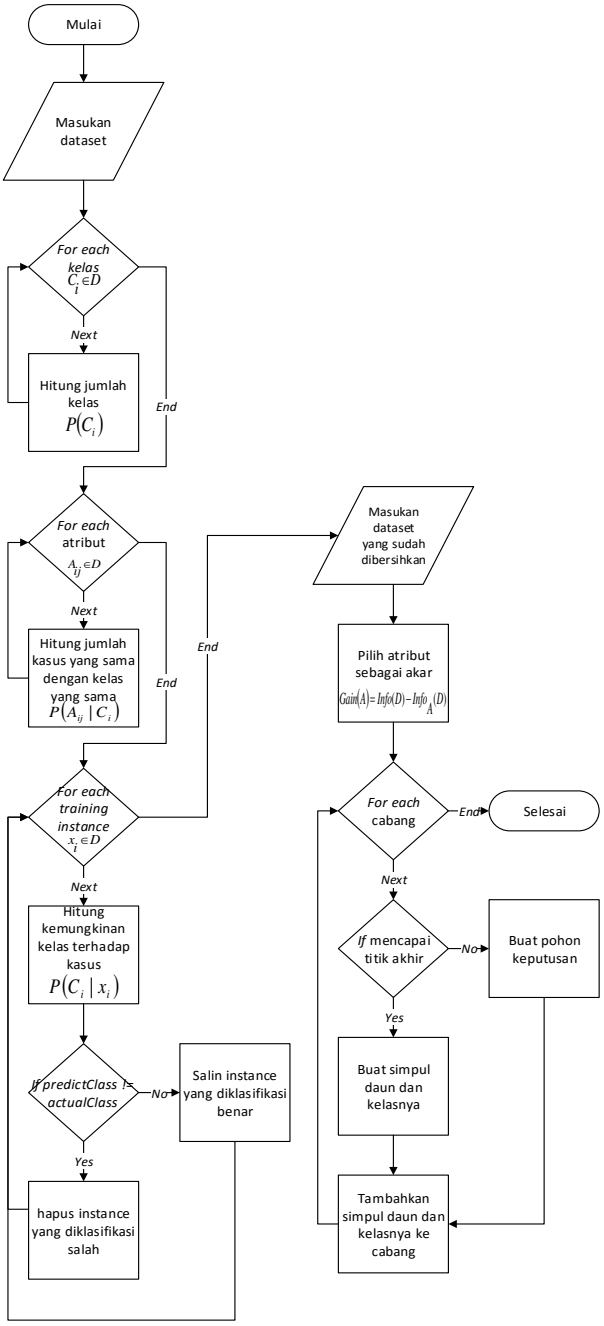


Gambar 2. Metode yang Diusulkan

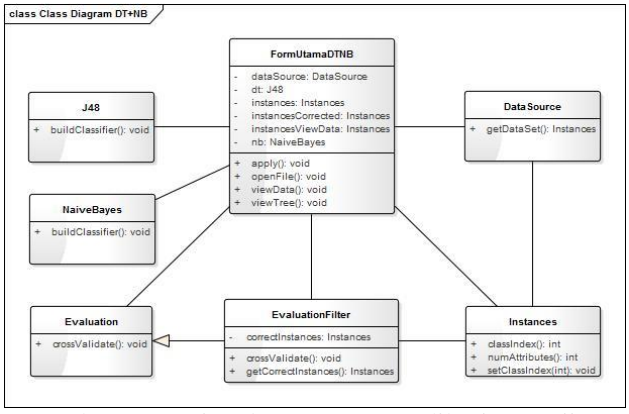
Dataset yang sudah dibersihkan dari *instance* yang salah klasifikasi kemudian dibagi menjadi data *training* dan data *testing* dengan menggunakan 10-*fold cross validation*, selanjutnya data *training* diproses dengan algoritma DT untuk menghasilkan pohon keputusan. Kemudian dengan model pohon keputusan tersebut dilakukan *testing*. Hasil validasi dari proses digunakan untuk mengukur kinerja algoritma dari metode yang diusulkan. Langkah-langkah pada penerapan algoritma pengklasifikasi NB untuk mengurangi *noise* pada klasifikasi multi kelas dengan DT ditunjukkan pada Gambar 3.

Proses eksperimen dan pengujian metode pada penelitian ini menggunakan antarmuka pengguna atau *user interface* (UI) dari aplikasi yang dikembangkan untuk mengukur kinerja metode yang diusulkan.

Aplikasi didesain menggunakan bahasa pemrograman Java dengan menggunakan *library* Weka. Rancangan dalam bentuk *class diagram* ditunjukkan pada Gambar 4 dan rancangan *form* utama UI aplikasi ditunjukkan pada Gambar 5.

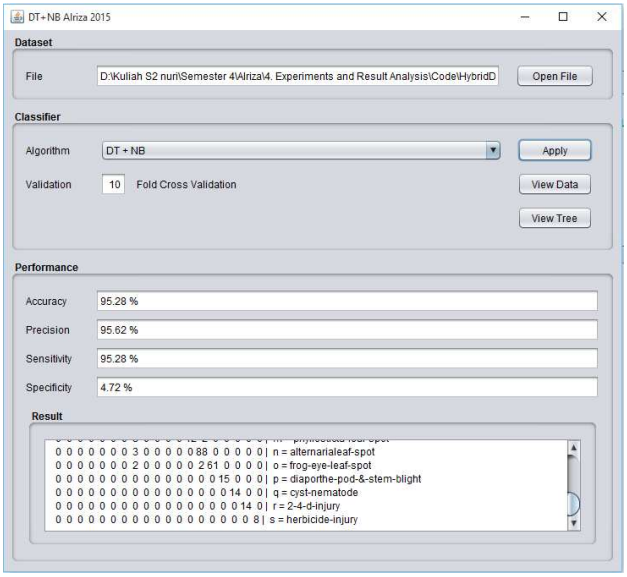


Gambar 3. *Flowchart* Metode yang Diusulkan



Gambar 4. Desain *Class Diagram* Aplikasi Pengujian

UI aplikasi memiliki tiga bagian yaitu *Dataset*, *Classifier*, dan *Performance*. Bagian *Dataset* memiliki tombol *Open File* berfungsi untuk memilih *file dataset*. Bagian *Classifier* memiliki *combo box Algorithm* untuk memilih algoritma yang akan digunakan, tombol *Apply* untuk memproses algoritma yang dipilih, tombol *View Data* untuk melihat *dataset* yang dipilih, tombol *View Tree* untuk melihat model pohon keputusan yang dihasilkan, *text box Validation* untuk menentukan jumlah *k-fold cross validation*. Bagian *Performance* memiliki *text box Accuracy*, *Precision*, *Sensitivity*, *Specificity*, dan *text area Result* yang menampilkan hasil kinerja klasifikasi.



Gambar 5. Aplikasi Pengujian

Dalam penelitian ini digunakan komputer untuk melakukan proses perhitungan terhadap metode yang diusulkan dengan spesifikasi komputer yang ditunjukkan pada Tabel 1.

Tabel 1. Spesifikasi Komputer

Processor	Intel Core i5-4210U 1,7 GHz
Memory	6 GB
Harddisk	1 TB
Operating System	Windows 10
Application	Java - Netbeans IDE 8.02

Pengukuran kinerja model menggunakan tabel *confusion matrix*. Pada tabel *confusion matrix* berisi nilai *false positive* (FP), *false negative* (FN), *true positive* (TP), dan *true negative* (TN). Kinerja yang diukur termasuk akurasi secara umum seperti *accuracy, precision, sensitivity,* dan *specificity*. Validasi yang dilakukan adalah dengan menggunakan 10-*fold cross validation* dimana *dataset* akan dibagi dalam dua segmen, data *training* dan data *testing* menjadi 10 bagian. Kinerja model akan dibandingkan antara algoritma *Decision Tree* (DT) + *Naive Bayes* (NB) dengan DT.

4 HASIL PENELITIAN

Eksperimen dilakukan dengan menggunakan laptop Dell 5000 *series* dengan *processor* Intel Core i5-4210U @ 1,7 GHz 2.40 GHz, *memory* 6 GB, *harddisk* 1 TB, dan menggunakan sistem operasi Windows 10 64-bit. Eksperimen ini juga menggunakan perangkat lunak Weka 3.6 untuk menganalisa penghitungan, dan menggunakan Netbeans IDE 8.02 dengan bahasa pemrograman Java dalam pengembangan aplikasi untuk menguji hasil perhitungan. Metode yang digunakan adalah dengan menerapkan algoritma pengklasifikasi *Naive Bayes* (NB) untuk mengurangi *noise* pada klasifikasi multi kelas dengan *Decision Tree* (DT). Algoritma pengklasifikasi NB digunakan untuk menemukan dan menghilangkan *instance* yang mengandung *noise*, sehingga akurasi pada klasifikasi yang dihasilkan oleh algoritma DT dapat meningkat.

Data yang digunakan pada penelitian ini adalah delapan *dataset* uji dari *University of California Irvine* (UCI) *machine learning repository* yang diperoleh melalui situs http://archive.ics.uci.edu/ml. *Dataset* tersebut digunakan oleh banyak peneliti untuk melakukan pengujian metode yang dibuat. *Dataset* tersebut juga bersifat publik dan dapat digunakan oleh siapa saja. *Dataset* yang digunakan dalam penelitian ini terdiri atas *dataset* yang memiliki dua kelas dan *dataset* yang memiliki lebih dari dua kelas atau multi kelas. Delapan *dataset* yang digunakan adalah sebagai berikut:

1. Data kanker payudara (*Breast Cancer*)
2. Data pasien diabetes (*Diabetes*)
3. Data klasifikasi kaca (*Glass*)
4. Data tanaman iris (*Iris*)
5. Data kacang kedelai (*Soybean*)
6. Data *voting* kongres di Amerika Serikat tahun 1984 (*Vote*)
7. Data segmentasi gambar (*Image Segmentation*)
8. Data permainan tic-tac-toe (*Tic-tac-toe*)

Dataset Breast Cancer adalah kumpulan data terkait klasifikasi penyakit kanker payudara, atribut yang dimiliki bertipe *nominal*, terdiri dari 286 *instances*, 10 atribut, dan 2 kelas.

Dataset Diabetes adalah kumpulan data terkait klasifikasi penyakit diabetes, atribut yang dimiliki bertipe *real*, terdiri dari 768 *instances*, 9 atribut, dan 2 kelas.

Dataset Glass adalah kumpulan data terkait klasifikasi tipe *glass* atau kaca, atribut yang dimiliki bertipe *real*, terdiri dari 214 *instances*, 10 atribut, dan 6 kelas.

Dataset Iris adalah kumpulan data terkait klasifikasi tanaman *iris*, atribut yang dimiliki bertipe *real*, terdiri dari 150 *instances*, 5 atribut, dan 3 kelas.

Dataset Soybean adalah kumpulan data terkait klasifikasi penyakit tanaman kedelai, atribut yang dimiliki bertipe *nominal*, terdiri dari 683 *instances*, 36 atribut, dan 19 kelas.

Dataset Vote adalah kumpulan data terkait klasifikasi pemilih dalam pemungutan suara di Amerika Serikat pada tahun 1984, atribut yang dimiliki bertipe *nominal*, terdiri dari 435 *instances*, 17 atribut, dan 2 kelas.

Dataset Image Segmentation adalah kumpulan data terkait klasifikasi gambar alam terbuka, atribut yang dimiliki bertipe *real*, terdiri dari 1500 *instances*, 20 atribut, dan 7 kelas.

Dataset Tic-tac-toe adalah kumpulan data terkait permainan bulat-silang, atribut yang dimiliki bertipe *nominal*, terdiri dari 958 *instances*, 10 atribut, dan 2 kelas. Tabel 2 menjelaskan spesifikasi dari delapan *dataset* UCI *machine learning repository*.

Tabel 2. Spesifikasi Delapan *Dataset* UCI *Machine Learning Repository*

Dataset	Jumlah atribut	Tipe atribut	Jumlah *instance*	Jumlah kelas
Breast cancer	10	*Nominal*	286	2
Diabetes	9	*Real*	768	2
Glass	10	*Real*	214	6
Iris	5	*Real*	150	3
Soybean	36	*Nominal*	683	19
Vote	17	*Nominal*	435	2
Image Segmentation	20	*Real*	1500	7
Tic-tac-toe	10	*Nominal*	958	2

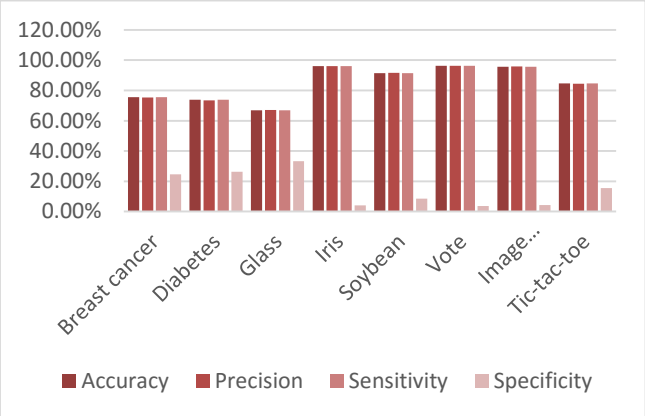
Eksperimen dalam penelitian dilakukan dalam dua metode, yaitu dengan menggunakan metode *Decision Tree* dan metode *Decision Tree* (DT) yang diintegrasikan dengan *Naive Bayes* (NB) atau DT+NB.

Pada eksperimen pertama ini percobaan dilakukan dengan menguji delapan *dataset* UCI *machine learning repository* menggunakan algoritma DT. Teknik validasi yang digunakan adalah 10-*fold cross validation*, dengan membagi *dataset* menjadi 10 bagian. Dari 10 bagian data tersebut, 9 bagian dijadikan data *training*, 1 bagian sisanya dijadikan data *testing*.

Berdasarkan hasil eksperimen, dilakukan perbandingan kinerja *Decision Tree* (DT) dengan *Decision Tree* dan *Naive Bayes* (DT + NB) untuk mengetahui algoritma klasifikasi yang terbaik. Pengukuran dilakukan dengan menguji delapan *dataset* dari UCI *machine learning repository* (*Breast Cancer, Diabetes, Glass, Iris, Soybean, Vote, Image Segmentation, Tic-tac-toe*). Hasil pengukuran algoritma klasifikasi dapat dilihat pada Tabel 3 dan grafik perbandingannya pada Gambar 6 untuk semua *dataset* dengan menggunakan algoritma DT, pada Tabel 4 dan Gambar 7 untuk semua *dataset* dengan menggunakan algoritma DT+NB.

Tabel 3. Hasil Pengukuran Algoritma Klasifikasi DT pada Semua Dataset Uji

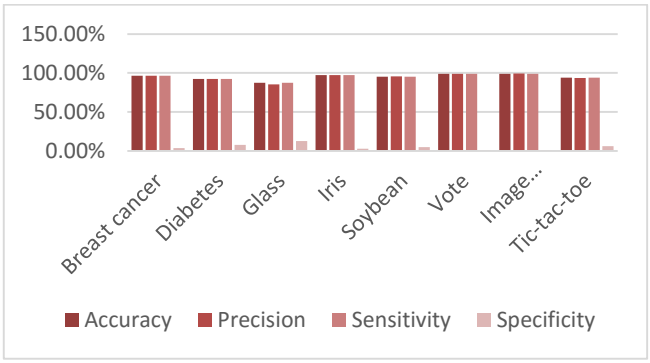
Dataset Training	*Accuracy*	*Precision*	*Sensitivity*	*Specificity*
Breast Cancer	75,52%	75,24%	75,52%	24,48%
Diabetes	73,83%	73,52%	73,83%	26,17%
Glass	66,82%	67,04%	66,82%	33,18%
Iris	96,00%	96,04%	96,00%	4,00%
Soybean	91,51%	91,65%	91,51%	8,49%
Vote	96,32%	96,32%	96,32%	3,68%
Image Segmentation	95,73%	95,78%	95,73%	4,27%
Tic-tac-toe	84,55%	84,49%	84,55%	15,45%



Gambar 6. Grafik Kinerja Algoritma Klasifikasi DT pada Semua Dataset Uji

Tabel 4. Hasil Pengukuran Algoritma Klasifikasi DT + NB pada Semua Dataset Uji

Dataset Training	*Accuracy*	*Precision*	*Sensitivity*	*Specificity*
Breast cancer	96,59%	96,63%	96,59%	3,41%
Diabetes	92,32%	92,34%	92,32%	7,68%
Glass	87,50%	85,46%	87,50%	12,50%
Iris	97,22%	97,25%	97,22%	2,78%
Soybean	95,28%	95,62%	95,28%	4,72%
Vote	98,98%	98,98%	98,98%	1,02%
Image Segmentation	99,10%	99,11%	99,10%	0,90%
Tic-tac-toe	93,85%	93,74%	93,85%	6,15%



Gambar 7. Grafik Kinerja Algoritma Klasifikasi DT + NB pada Semua Dataset Uji

Selanjutnya dilakukan Uji beda dengan metode statistik yang digunakan untuk menguji hipotesis pada algoritma DT dengan algoritma DT + NB.

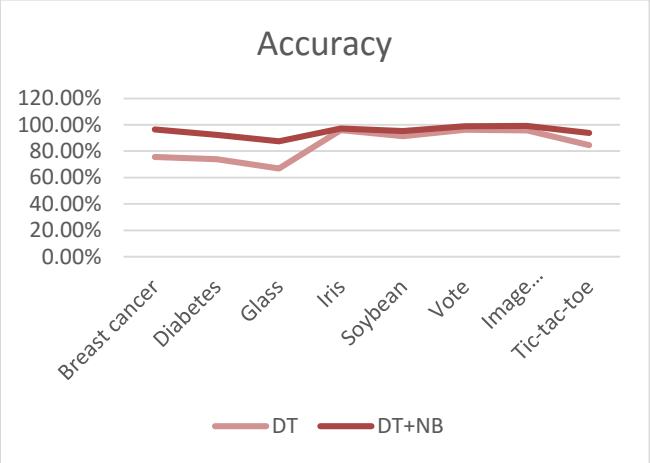
H_0: Tidak ada perbedaan antara nilai rata-rata *accuracy* DT dengan DT + NB

H_1: Ada perbedaan antara nilai rata-rata *accuracy* DT dengan DT + NB

Perbedaan nilai *accuracy* antara DT dengan DT + NB disajikan dalam Tabel 5 dan Gambar 8.

Tabel 5. Perbandingan *Accuracy* DT dengan DT + NB

Dataset Training	DT	DT + NB
Breast Cancer	75,52%	96,59%
Diabetes	73,83%	92,32%
Glass	66,82%	87,50%
Iris	96,00%	97,22%
Soybean	91,51%	95,28%
Vote	96,32%	98,98%
Image Segmentation	95,73%	99,10%
Tic-tac-toe	84,55%	93,85%



Gambar 8. Grafik Perbandingan *Accuracy* DT dengan DT + NB

Hasil analisis dengan menggunakan uji *t-Test Paired Two Sample for Means* disajikan dalam Tabel 6.

Tabel 6. Hasil Uji Beda Statistik *Accuracy* DT dengan DT + NB

	DT	*DT + NB*
Mean	0,850362	**0,95104**
Variance	0,013599	0,001497
Observations	8	8
Pearson Correlation	0,845275	
Hypothesized Mean Difference	0	
Df	7	
t Stat	-3,29507	
P(T<=t) one-tail	0,006605	
t Critical one-tail	1,894579	
P(T<=t) two-tail	**0,01321**	
t Critical two-tail	2,364624	

Pada Tabel 6 dapat dilihat bahwa nilai rata-rata *accuracy* dari algoritma DT + NB lebih tinggi dibandingkan algoritma DT sebesar 0,95104. Dalam uji beda statistik nilai *alpha* ditentukan sebesar 0,05, jika nilai p lebih kecil dibandingkan *alpha* ($p < 0,05$) maka H_0 ditolak dan H_1 diterima sehingga disimpulkan ada perbedaan yang signifikan antara algoritma yang dibandingkan, namun bila nilai p lebih besar dibanding *alpha* ($p > 0,05$) maka H_0 diterima dan H_1 ditolak sehingga disimpulkan tidak ada perbedaan yang signifikan antara algoritma yang dibandingkan. Pada Tabel 4.36 dapat diketahui bahwa nilai P(T<=t) adalah 0,01321, ini menunjukkan bahwa nilai p lebih kecil daripada nilai *alpha* (0,01321 < 0,05) sehingga hipotesis H_0 ditolak dan H_1 diterima. Dengan demikian dapat disimpulkan bahwa ada perbedaan yang signifikan antara algoritma DT dengan DT + NB.

5 KESIMPULAN

Dalam penelitian ini algoritma pengklasifikasi *Naive Bayes* (NB) digunakan untuk menemukan dan menghilangkan *instance* yang mengandung *noise*, sehingga akurasi pada klasifikasi yang dihasilkan oleh algoritma *Decision Tree* (DT) dapat meningkat. Pengujian dilakukan pada delapan *dataset* dari UCI *machine learning repository* dengan menggunakan algoritma yang diusulkan dan algoritma DT. *Dataset* yang digunakan dalam pengujian terdiri atas *dataset* yang memiliki dua kelas dan *dataset* yang memiliki lebih dari dua kelas atau multi kelas.

Berdasarkan hasil eksperimen dan evaluasi pada penelitian ini, secara umum dapat disimpulkan bahwa penerapan algoritma pengklasifikasi NB dapat mengurangi data *noise* pada *dataset* berukuran besar dan memiliki banyak kelas atau multi kelas sehingga akurasi klasifikasi algoritma DT dapat meningkat. Hasil akurasi yang didapat menunjukkan bahwa metode yang diusulkan DT+NB lebih unggul dari metode DT, dengan nilai akurasi untuk masing-masing *dataset* uji seperti *Breast Cancer* 96,59% (meningkat 21,06%), *Diabetes* 92,32% (meningkat 18,49%), *Glass* 87,50% (meningkat 20,68%), *Iris* 97,22% (meningkat 1,22%), *Soybean* 95,28% (meningkat 3,77%), *Vote* 98,98% (meningkat 2,66%), *Image Segmentation* 99,10% (meningkat 3,36%), dan *Tic-tac-toe* 93,85% (meningkat 9,30%). Perbandingan nilai akurasi dilakukan dengan uji t atau *t-Test* antara metode DT dengan metode yang diusulkan DT + NB untuk mendapatkan nilai perbedaan akurasi signifikan antara kedua metode tersebut. Dari hasil perbandingan didapatkan nilai P(T<=t) adalah 0,01321, ini menunjukkan bahwa nilai p lebih kecil daripada nilai *alpha* (0,01321 < 0,05). Dengan demikian dapat disimpulkan bahwa ada perbedaan akurasi yang signifikan antara metode DT dengan DT + NB.

REFERENSI

Aggarwal, C. C. (2015). *Data Mining, The Textbook*. Springer Berlin Heidelberg.

Aitkenhead, M. J. (2008). A co-evolving decision tree classification method. *Expert Systems with Applications*, *34*(1), 18–25.

Aviad, B., & Roy, G. (2011). Classification by clustering decision tree-like classifier based on adjusted clusters. *Expert Systems with Applications*, *38*(7), 8220–8228.

Balamurugan, S. A. A., & Rajaram, R. (2009). Effective solution for unhandled exception in decision tree induction algorithms. *Expert Systems with Applications*, *36*(10), 12113–12119.

Berndtsson, M., Hansson, J., Olsson, B., & Lundell, B. (2008). *Thesis Guide - A Guide for Students in Computer Science and Information Systems (2nd ed)*. Springer-Verlag.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees*. *Chapman and Hall/CRC* (1st ed., Vol. 19). Chapman and Hall/CRC.

Bujlow, T., Riaz, T., & Pedersen, J. M. (2012). A method for classification of network traffic based on C5.0 machine learning algorithm. *2012 International Conference on Computing, Networking and Communications, ICNC'12*, 237–241.

Chandra, B., & Paul Varghese, P. (2009). Fuzzifying Gini Index based decision trees. *Expert Systems with Applications*, *36*(4), 8549–8559.

Chen, Y. L., & Hung, L. T. H. (2009). Using decision trees to summarize associative classification rules. *Expert Systems with Applications*, *36*, 2338–2351.

Dawson, C. W. (2009). *Projects in Computing and Information Systems A Student's Guide (2nd ed)*. *Great Britain: Pearson Education*.

Demsar, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. *The Journal of Machine Learning Research*, *7*, 1–30.

Farid, D. M., & Rahman, M. Z. (2010). Anomaly network intrusion detection based on improved self adaptive Bayesian algorithm. *Journal of Computers*, *5*(1), 23–31.

Farid, D. M., Rahman, M. Z., & Rahman, C. M. (2011). Adaptive Intrusion Detection based on Boosting and Naive Bayesian Classifier. *International Journal of Computer Applications*, *24*(3), 12–19.

Farid, D. M., Zhang, L., Hossain, A., Rahman, C. M., Strachan, R., Sexton, G., & Dahal, K. (2013). An adaptive ensemble classifier for mining concept drifting data streams. *Expert Systems with Applications*, *40*(15), 5895–5906.

Franco-Arcega, A., Carrasco-Ochoa, J. a., Sanchez-Diaz, G., & Martinez-Trinidad, J. F. (2011). Decision tree induction using a fast splitting attribute selection for large datasets. *Expert Systems with Applications*, *38*(11), 14290–14300.

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques*.

Jamain, A., & Hand, D. J. (2008). Mining Supervised Classification Performance Studies: A Meta-Analytic Investigation. *Journal of Classification*, *25*(1), 87–112.

Larose Daniel T. (2005). *Discovering Knowledge in Data: An Introduction to Data Mining*. Wiley Interscience.

Lee, L. H., & Isa, D. (2010). Automatically computed document dependent weighting factor facility for Naïve Bayes classification. *Expert Systems with Applications*, *37*(12), 8471–8478.

Liao, S. H., Chu, P. H., & Hsiao, P. Y. (2012). Data mining techniques and applications - A decade review from 2000 to 2011. *Expert Systems with Applications*, *39*(12), 11303–11311.

Loh, W.-Y., & Shih, Y.-S. (1997). Split Selection Methods for Classification Trees. *Statistica Sinica*, *7*(4), 815–840.

Maimon, O., & Rokach, L. (2010). *Data Mining and Knowledge Discovery Handbook*. *Data Mining and Knowledge Discovery Handbook* (2nd ed.). New York: Springer-Verlag.

Ngai, E. W. T., Xiu, L., & Chau, D. C. K. (2009). Application of data mining techniques in customer relationship management: A literature review and classification. *Expert Systems with Applications*, *36*, 2592–2602.

Polat, K., & Gunes, S. (2009). A novel hybrid intelligent method based on C4.5 decision tree classifier and one-against-all approach for multi-class classification problems. *Expert Systems with Applications*, *36*, 1587–1592.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*(1), 81–106.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. California: Morgan Kaufmann.

Safavian, S. R., & Landgrebe, D. (1991). A Survey of Decision Tree Classifier Methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, *21*(3).

Turney, P. (1995). Cost-Sensitive Classification : Empirical Evaluation of a Hybrid Genetic Decision Tree Induction Algorithm. *Journal of Artificial Intelligence Research*, *2*, 369–409.

Utgoff, P. E. (1989). Incremental Induction of Decision Trees. *Machine Learning*, *4*(2), 161–186.

Witten, I. H., Eibe, F., & Hall, M. A. (2011). *Data mining : practical machine learning tools and techniques.—3rd ed. Morgan Kaufmann* (3rd ed.). Morgan Kaufmann.

BIOGRAFI PENULIS

Al Riza Khadafy. Memperoleh gelar S.Kom pada jurusan Sistem Informasi dari STMIK Nusa Mandiri, Jakarta dan gelar M.Kom pada jurusan Ilmu Komputer dari Pascasarjana STMIK Nusa Mandiri Jakarta. Bekerja sebagai staff IT di perusahaan swasta di Jakarta. Minat penelitian pada saat ini meliputi bidang *data mining* dan *machine learning*.

Romi Satria Wahono. Memperoleh gelar B.Eng dan M.Eng pada bidang ilmu komputer di Saitama University Japan, dan Ph.D pada bidang software engineering di Universiti Teknikal Malaysia Melaka. Pengajar dan peneliti di Fakultas Ilmu Komputer, Universitas Dian Nuswantoro. Pendiri dan CEO PT Brainmatics, perusahaan yang bergerak di bidang pengembangan software. Minat penelitian pada bidang software engineering dan machine learning. Profesional member dari asosiai ilmiah ACM, PMI dan IEEE Computer Society.