Prediksi Cacat Perangkat Lunak Dengan Optimasi Naive Bayes Menggunakan Pemilihan Fitur Gain Ratio

Muhammad Sonhaji Akbar, Siti Rochimah
Teknik Informatika, Institut Teknologi Sepuluh Nopember
Jl. Raya ITS, Keputih, Sukolilo, Kota Surabaya, Jawa Timur 031-5994251
e-mail: : mson.akbar@gmail.com, siti@its-sby.edu

Abstrak

Dalam prediksi cacat perangkat lunak, terjadinya kesalahan prediksi cacat perangkat lunak merupakan hal yang sangat fatal karena data yang salah prediksi dapat menimbulkan pengaruh terhadap perangkat lunak itu sendiri. Kurang optimalnya metode prediksi yang digunakan. Masih terdapat kesalahan dalam memprediksi cacat perangkat lunak. Dalam metode Naive Bayes juga masih terdapat kekurangan ketika terjadi kesalahan klasifikasi. Kesalahan klasifikasi ini dapat memperlambat proses prediksi cacat perangkat lunak. Dibutuhkan metode yang dapat mengatasi kesalahan klasifikasi ini. Dalam penelitian ini disusulkan optimasi metode Naive Bayes menggunakan Gain Ratio. Pemilihan fitur menggunakan Gain Ratio pada Naïve Bayes dapat mengurangi dampak kegagalan prediksi. Penggunaan Gain Ratio dapat meningkatkan performa prediksi. Penghitungan Gain Ratio dapat dirumuskan yaitu dari setiap atribut Gain Ratio dikali jumlah data n kemudian dibagi dengan rata-rata Gain Ratio semua atribut. Atribut dari Gain Ratio sendiri merupakan hasil bagi dari Mutual Information dan Entropy. Mutual Information (MI) merupakan nilai ukur yang menyatakan keterikatan atau ketergantungan antara dua variabel atau lebih. Selain MI, Entropy digunakan sebagai pembagi dari MI yang digunakan untuk menentukan atribut mana yang terbaik atau optimal. Maka dari itu penghitungan Gain Ratio adalah hasil dari penghitungan Mutual Information dibagi dengan hasil penghitungan Entropy Penghitungan Gain Ratio. Hasil penelitian menunjukkan akurasi sebesar 87,55% untuk metode usulan dan 85,34% untuk metode Naïve Bayes biasa.

Kata kunci: *Prediksi, Klasifikiasi, Cacat Perangkat Lunak, Naïve Bayes, Gain Ratio*

Abstract

In the prediction of software defects, the occurrence of software defects prediction error is very serious because the data is wrong predictions can be impacting on the software itself. Less optimal prediction methods used. Still there are errors in predicting software defects. In a Naive Bayes method is still a shortfall in the event of misclassification. Misclassification can slow the process of software defects prediction. It takes a method to resolve this misclassification. In this study be followed optimization Naive Bayes method using Gain Ratio. Gain Ratio feature selection using the Naïve Bayes can reduce the impact of failure prediction. Use of Gain Ratio can improve performance predictions. Gain Ratio Calculations can be formulated that of each attribute Gain Ratio multiplied by the number of data n is then divided by the average Gain Ratio of all attributes. Attributes of Gain Ratio itself is the quotient of the Mutual Information and Entropy. Mutual Information (MI) is a measure that states the value of attachment or dependency between two or more variables. Besides MI, Entropy is used as a divisor of MI is used to determine which attributes best or optimal. Thus the calculation of Gain Ratio is the result of the calculation of the Mutual Information shared with the calculation results Entropy Calculations Gain Ratio. The results showed an accuracy of 87.55% for the proposed method and 85.34% for the Naïve Bayes methods usual.

Keywords*: Prediction, Classification, Defect, Software, Naïve Bayes, Gain Ratio.*

1. Pendahuluan

Mengembangkan perangkat lunak yang berkualitas dan kompleks membutuhkan biaya yang tinggi. Dibutuhkan cara yang efektif untuk meminimalkan biaya dan usaha dalam membangun perangkat lunak. Salah satu cara untuk mengembangkan perangkat lunak berkualitas adalah dengan meminimalkan terjadinya cacat ketika perangkat lunak telah dijalankan menggunakan teknik prediksi cacat perangkat lunak. Prediksi cacat perangkat lunak dapat mendeteksi modul terkecil perangkat lunak yang memiliki kecenderungan cacat.

Terdapat beberapa penelitian yang telah dikembangkan untuk membangun metode prediksi cacat perangkat lunak tetapi masih kurang optimal seperti Relational Association Rule Mining (Czibula, 2014)butuh mencari kombinasi data yang sering muncul sehingga memerlukan waktu komputasi yang lama. Selain itu metode Support Vector Machine dapat dikatakan baik tergantung pada dataset yang digunakan, ketika menggunakan data dengan dua kelas cocok tapi tidak cocok ketika data yang diolah besar [14]. Metode lain yaitu Cost Sensitive Neural Network dapat digunakan pada data apa saja, tetapi karena data butuh pelatihan terlebih dahulu maka komputasi yang dihasilkan akan lebih lama. Terdapat satu lagi metode yaitu Neural Network (NN) yang digabungkan dengan Particle swarm optimization (PSO) dan Association Rule, dalam metode ini PSO mudah diimplementasikan akan tetapi pengggunakan PSO ini hanya cocok digunakan untuk metode NN. Beberapa masalah dalam metode-metode tersebut dapat diatasi menggunakan Naive Bayes dengan Pembobotan [4], karena tidak membutuhkan waktu komputasi yang lama, lebih efektif, efisien, dapat mereduksi atribut, stabil dan akurasi dapat meningkat. Salah satu pembobotan menggunakan Gain Ratio [6], Gain Ratio dapat memperbaiki data yang tidak stabil, cocok untuk data numeric dua kelas, sederhana sehingga komputasi lebih cepat. Terdapat juga metode Naive Bayes dengan optimasi Gain Ratio [7] tetapi digunakan dalam prediksi teks berita.

Naive Bayes memiliki beberapa kelebihan [8], yaitu algoritma yang sederhana, lebih cepat dalam penghitungan dan berakurasi tinggi. Akan tetapi, pada metode Naive Bayes juga memiliki kelemahan [7] dimana sebuah probabilitas tidak bisa mengukur seberapa besar tingkat keakuratan sebuah prediksi. Dengan kata lain, jika sebuah probabilitas tidak dapat merepresentasikan sebuah data maka prediksi yang dihasilkan kurang akurat. Selain itu, terdapat permasalahan data yang sering muncul pada kelas lain dan muncul juga pada kelas yang diuji mengakibatkan kesalahan prediksi. Hal ini yang menyebabkan metode Naive Bayes masih belum optimal.

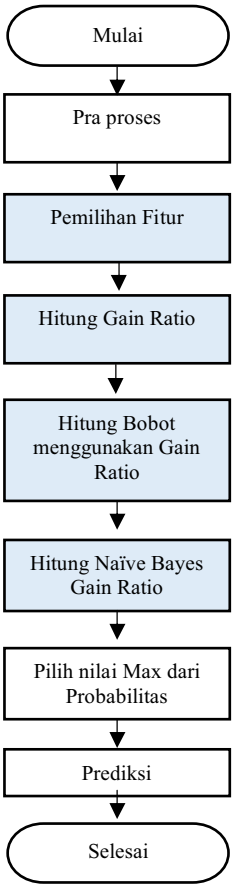
Penelitian ini mengusulkan metode optimasi Naive Bayes mennggunakan Gain Ratio untuk meningkatkan meningkatkan akurasi prediksi cacat perangkat lunak. Menurut Zaidi [9], pembobotan atribut pada Naive Bayes dapat mengurangi dampak kegagalan prediksi. Salah satu metode yang dapat digunakan untuk pembobotan Naive Bayes yaitu Gain Ratio. Gain Ratio digunakan untuk memilih atribut terbaik di antara seluruh atribut pada metode prediksi. Penggunaan Gain Ratio diharapkan dapat meningkatkan akurasi prediksi. Atribut dari Gain Ratio sendiri merupakan hasil bagi dari Mutual Information dan Entropy.

2. Tinjauan Pustaka/ State of the Art

Cacat perangkat lunak *(Software Defect)* didefinisikan sebagai cacat pada perangkat lunak seperti cacat pada dokumentasi, pada kode program, pada desain dan hal – hal lain yang menyebabkan kegagalan perangkat lunak. Penelitian ini hanya melakukan prediksi cacat perangkat lunak pada modul atau kode programmnya saja. Cacat perangkat lunak dapat muncul pada berbagai tahap proses pengembangan perangkat lunak [10]. Cacat perangkat lunak merupakan faktor penting yang mempengaruhi kualitas perangkat lunak. Kualitas perangkat lunak dapat ditingkatkan dengan mencegah munculnya cacat perangkat lunak melalui perbaikan aksi yang mungkin menghasilkan cacat perangkat lunak pada proses pengembangan perangkat lunak [11]. Dalam melakukan prediksi, metode Naive Bayes dengan Gain Ratio ini memperbaiki data yang tidak stabil, cocok digunakan untuk data numerik dengan dua kelas, sederhana sehinggga waktu komputasi lebih cepat. kombinasi data yang paling sering muncul, mendefinisikan kondisi data yang telah dikombinasi sehingga membutuhkan waktu komputasi yang lebih lama. Selain itu metode ini lebih efektif, efisien karena dapat mereduksi fitur atau atribut yang ada tetapi tetap stabil sehingga akurasi yang didapatkan dapat meningkat dan waktu komputasi lebih cepat [4][6].

3. Metode Penelitian

Metode yang digunakan dalam penelitian ini yaitu menggunakan gabungan metode dalam Prediksi Cacat Perangkat Lunak yaitu Naive Bayes dan Gain Ratio. Berikut metodenya:



Gambar 1. Alur metode

Berdasarkan Gambar 1 Langkah awal penelitian ini adalah melakukakan pra proses dengan melakukakn pemilihan fitur. Menurut Hilden and Bjerregaard, Ferreira dan Hall, pembobotan atribut kelas dapat meningkatkan pengaruh prediksi. Dengan memperhitungkan bobot atribut terhadap kelas, maka yang men-jadi dasar ketepatan klasisifikasi bukan hanya probabilitas melainkan juga dari bobot setiap atribut ter-hadap kelas. Pembobotan Naive Bayes dihitung dengan cara menambahkan bobot wi pada setiap atribut. Sehingga didapatkan rumus untuk pembobotan Naive Bayes dituliskan pada Persamaan (1).

$$P(y,x) = P(y)\prod_{i=1}^{a} C * P(x_i \mid y)^{w_i} \quad (1)$$

Pembobotan dapat dirumuskan menggunakan Gain Ratio. Dimana dari setiap atribut Gain Ratio dikali jumlah data n kemudian dibagi dengan rata-rata Gain Ratio semua atribut. Atribut dari Gain Ratio sendiri merupakan hasil bagi dari Mutual Information dan Entropy. Mutual Information (MI) merupakan nilai ukur yang menyatakan keterikatan atau ketergantungan antara dua variabel atau lebih. Unit pengukur yang umum digunakan untuk menghitung MI adalah bit, sehingga menggunakan logaritma (log) basis 2. Secara formal, MI digunakan antara 2 variabel A dan B. Selain MI, Entropy digunakan sebagai pembagi dari MI yang digunakan untuk menentukan atribut mana yang terbaik atau optimal.

Sebelum mendapatkan nilai Gain Ratio dilakukan pencarian nilai Entropy (E). Entropy digunakan untuk menentukan seberapa informatif sebuah masukan atribut untuk menghasilkan keluaran atribut. Maka dari itu penghitungan Gain Ratio adalah hasil dari penghitungan Mutual Information dibagi dengan hasil penghitungan Entropy Penghitungan Gain Ratio.

Proses penghitungan Naive Bayes menggunakan Gain Ratio dibagi menjadi dua tahap. Tahap pertama adalah proses pelatihan. Pada proses pelatihan diambil data latih kemudian dilakukan pra proses. Setelah itu hitung peluang data per-kategori dan hitung peluang kategori kelas. Kemudian dicari nilai Gain Ratio.

Tahap kedua adalah proses pelatihan. Pada proses pengujian diambil data uji kemudian dilakukan pra proses. Setelah itu ambil nilai Gain Ratio tiap data dan kategori. Setelah itu, dilakukan proses perankingan data sebanyak R (jumlah data yang ditentukan). Dari data sebanyak R yang diambil dilakukakn proses penghitungan Gain Ratio. Kemudian dicari nilai bobot Naive Bayes menggunakan Persamaan 10. Setelah itu dicari nilai makasimum dan minimum kemungkinan, lalu dapat ditentukan hasil prediksinya.

3.1. Dataset

Dataset yang digunakan dalam penelitian ini adalah histori cacat perangkat lunak yang berasal dari NASA Metric Data Program (MDP). Dataset perangkat lunak yang pernah dibuat akan digunakan untuk memprediksi terdapat cacat atau tidak pada perangkat lunak sebelum dibuat yang mirip dengan sebe-lumnya. Di dalam dataset NASA MDP terdapat beberapa studi kasus cacat perangkat lunak. Berikut contoh data set dan atributnya:

1. Jenis dataset : CM1/software defect prediction
2. Jumlah instan : 498
3. Jumlah atribut : 22 (5 different lines of code measure, 3 McCabe metrics, 4 base Halstead measures, 8 derived Halstead measures, a branch-count, and 1 goal field)
4. Informasi Atribut :
   1. loc : numeric % McCabe's line count of code
   2. v(g) : numeric % McCabe "cyclomatic complexity"
   3. ev(g) : numeric % McCabe "essential complexity"
   4. iv(g) : numeric % McCabe "design complexity"
   5. n : numeric % Halstead total operators + operands
   6. v : numeric % Halstead "volume"
   7. l : numeric % Halstead "program length"
   8. d : numeric % Halstead "difficulty"
   9. i : numeric % Halstead "intelligence"
   10. e : numeric % Halstead "effort"
   11. b : numeric % Halstead
   12. t : numeric % Halstead's time estimator
   13. lOCode : numeric % Halstead's line count
   14. lOComment : numeric % Halstead's count of lines of comments
   15. lOBlank : numeric % Halstead's count of blank lines
   16. lOCodeAndComment : numeric
   17. uniq_Op : numeric % unique operators
   18. uniq_Opnd : numeric % unique operands
   19. total_Op : numeric % total operators
   20. total_Opnd : numeric % total operands
   21. branchCount : numeric % of the flow graph
   22. defects : {false,true} % module has/has not one or more reported defects

3.2. Preprocessing

Pemilihan dilakukan sebelum data diolah agar komputasi yang dilakukan cepat. Pemilihan fitur ini menggunakan perangkingan Gain Ratio. Berikut hasil pengolahan pemilihan fiturnya. Pemilihan fitur berikut dilakukan menggunakan dataset CM1, JM1, PC1, KC1, KC2.

Tabel 1. Pemilihan Fitur Dataset CM1

No.		Ranked attributes:	
1	**0.0574**	**14**	**lOComment**
2	**0.0531**	**17**	**uniq_Op**
3	**0.0526**	**18**	**uniq_Opnd**
4	**0.052**	**11**	**b**
5	**0.0515**	**1**	**loc**
6	0.0466	9	i
7	0.0455	6	v
8	0.0453	4	iv(g)
9	0.0447	5	n
10	0.0437	19	total_Op
11	0.0408	15	lOBlank
12	0.0393	10	e
13	0.0393	12	t
14	0.0363	20	total_Opnd
15	0.0319	8	d
16	0.0306	7	l
17	0.0282	21	branchCount
18	0.0275	2	v(g)
19	0.0203	13	lOCode
20	0	3	ev(g)
21	0	16	locCodeAndComment

14,17,18,11,1,9,6,4,5,19,15,10,12,20,8,7,21,2,13, 3,16 : 21	**Selected attributes**

Tabel 2. Pemilihan Fitur Dataset JM1

No.		Ranked attributes:	
1	**0.0264**	**1**	**loc**
2	**0.0228**	**2**	**v(g)**
3	**0.0226**	**21**	**branchCount lOCode**
4	**0.0222**	**13**	**n**
5	**0.0221**	**5**	iv(g)
6	0.0219	4	e
7	0.0214	10	locCodeAndComment
8	0.0214	16	t
9	0.0213	12	v
10	0.0213	6	ev(g)
11	0.0209	3	i
12	0.0207	9	total_Op
13	0.0202	19	uniq_Opnd
14	0.0197	18	total_Opnd l
15	0.019	20	uniq_Op
16	0.0184	17	b
17	0.0183	11	lOComment
18	0.0175	14	lOBlank
19	0.0174	15	d
20	0.0162	8	l
21	0.0151	7	

1,2,21,13,5,4,10,16,12,6,3,9,19,18,20,17,11,14, 15,8,7 : 21	**Selected attributes**

Hasil dari pemilihan fitur ini diambil lima dengan bobot yang paling besar karena semakin besar bobotnya semakin besar pengaruh terhadap hasil prediksi cacat perangkat lunak.

3.3. Naive Bayes dengan Gain Ratio

Naive Bayes adalah metode yang digunakan dalam statistika untuk menghitung peluang dari suatu hipotesis, Naive Bayes menghitung peluang suatu kelas berdasarkan pada atribut yang dimiliki dan menentukan kelas yang memiliki probabilitas paling tinggi. Naive Bayes mengprediksikan kelas berdasarkan pada probabilitas sederhana dengan mangasumsikan bahwa setiap atribut dalam data tersebut bersifat saling terpisah. Metode Naive Bayes merupakan salah satu metode yang banyak digunakan berdasarkan beberapa sifatnya yang sederhana, metode Naive Bayes mengprediksikan data berdasarkan probabilitas P atribut x dari setiap kelas y data. Pada model probablitas setiap kelas k dan jumlah atribut a yang dapat dituliskan seperti Persamaan (2) [2] berikut.

$$\boldsymbol{P(y_k \mid x_1, x_2, \ldots x_a)} \quad \textbf{... (2)}$$

Penghitungan Naive Bayes yaitu probabilitas dari kemunculan dokumen xa pada kategori kelas yk P(xa|yk), dikali dengan probabilitas kategori kelas P(yk). Dari hasil kali tersebut kemudian dilakuan pembagian terhadap probabilitas kemunculan dokumen P(xa). Sehingga didapatkan rumus penghitungan Naive Bayes dituliskan pada Persamaan (3) [2].

$$\boldsymbol{P(y_k \mid x_a) = \frac{P(y_k)P(x_a \mid y_k)}{P(x_a)}} \quad \textbf{... (3)}$$

Kemudian dilakukan proses pemilihan kelas yang optimal maka dipilih nilai peluang terbesar dari setiap probabilitas kelas yang ada. Sehingga didapatkan rumus untuk memilih nilai terbesar pada Persamaan (4) [11].

$$\boldsymbol{y(x_i) = \arg\max P(y)\prod_{i=1}^{a} P(x_i \mid y)} \quad \textbf{... (4)}$$

Menurut Hilden and Bjerregaard, Ferreira dan Hall, pembobotan atribut kelas dapat meningkatkan pengaruh prediksi [8][9][10]. Dengan memperhi-tungkan bobot atribut terhadap kelas, maka yang menjadi dasar ketepatan klasisifikasi bukan hanya probabilitas melainkan juga dari bobot setiap atribut terhadap kelas. Pembobotan Naive Bayes dihitung dengan cara menambahkan bobot wi pada setiap atribut. Sehingga didapatkan rumus untuk pembo-botan Naive Bayes dituliskan pada Persamaan (5).

$$P(y,x) = P(y)\prod_{i=1}^{a} C * P(x_i \mid y)^{w_i} \quad \textbf{...(5)}$$

Pembobotan dapat dirumuskan menggunakan Gain Ratio [11]. Dimana dari setiap atribut Gain Ratio dikali jumlah data n kemudian dibagi dengan rata-rata Gain Ratio semua atribut.

$$\boldsymbol{w_i = \frac{GainRatio(i)}{\frac{1}{a}\sum_{i=1}^{a} GainRatio(i)}} \quad \textbf{...(6)}$$

Atribut dari Gain Ratio sendiri merupakan hasil bagi dari Mutual Information dan Entropy. Mutual Information (MI) merupakan nilai ukur yang menyatakan keterikatan atau ketergantungan antara dua variabel atau lebih. Unit pengukur yang umum digunakan untuk menghitung MI adalah bit, sehingga menggunakan logaritma (log) basis 2. Secara formal, MI digunakan antara 2 variabel A dan B yang didefinisikan oleh Kulback dan Leibler [12], Rényi [13]. Selain MI, Entropy digunakan sebagai pembagi dari MI yang digunakan untuk

menentukan atribut mana yang terbaik atau optimal. Penghitungan Mutual Information dituliskan pada Persamaan 7 [12][13].

$$MI(x_i, y) = \sum_y \sum_{x1} P(x_1, y) \log \frac{P(x_1, y)}{P(x_1)P(y)} \quad \text{...(7)}$$

Sebelum mendapatkan nilai Gain Ratio dilakukan pencarian nilai Entropy E. Entropy digunakan untuk menentukan seberapa informatif sebuah masukan atribut untuk menghasilkan keluaran atribut. Penghitungan Entropy dengan menjumlahkan probabilitas dituliskan pada Persamaan 8.

$$E(x_i) = \sum_{x1} P(x_1) \log \frac{1}{P(x_1)} \quad \text{...(8)}$$

Maka dari itu penghitungan Gain Ratio adalah hasil dari penghitungan Mutual Information dibagi dengan hasil penghitungan Entropy Penghitungan Gain Ratio dituliskan pada Persamaan 9.

$$GainRatio(i) = \frac{MI(x_i, y)}{E(x_i)} = \frac{\sum_y \sum_{x1} P(x_1, y) \log \frac{P(x_1, y)}{P(x_1)P(y)}}{\sum_{x1} P(x_1) \log \frac{1}{P(x_1)}} \quad \text{...(9)}$$

3.4. Metode Evaluasi

Tahap evaluasi bertujuan untuk mengetahui tingkat akurasi dari hasil penggunaan metode Weighted Naive Bayes. Dari evaluasi akan tersedia informasi mengenai seberapa besar akurasi yang telah dicapai. Pada proses pengujian dikenal sebagai Matriks Confusion yang merepresentasikan kebenaran dari sebuah prediksi. Tabel Matriks Confusion dapat dilihat pada Tabel 1.

Tabel 1. Metode Evaluasi

		Hasil Prediksi	
		+	-
Kenyataan	+	***True Positive***	***False Positive***
	-	***False Negative***	***True Negative***

- True Positive (TP) menunjukkan bahwa dokumen yang termasuk dalam hasil penge lompokkan oleh sistem memang merupakan anggota kelas.
- False Positive (FP) menunjukkan bahwa dokumen yang termasuk dalam hasil penge-lompokkan oleh sistem ternyata seharusnya bukan merupakan anggota kelas.
- False Negative (FN) menunjukkan bahwa dokumen yang tidak termasuk dalam hasil pengelompokkan oleh sistem ternyata seharusnya merupakan anggota kelas.
- True Negative (TN) menunjukkan bahwa dokumen yang tidak termasuk dalam hasil pengelompokkan oleh sistem ternyata seha-rusnya bukan merupakan anggota kelas.

Akurasi menunjukkan kedekatan nilai hasil pen-gukuran dengan nilai sebenarnya. Untuk menen-tukan tingkat akurasi perlu diketahui nilai sebenarnya dari parameter yang diukur [16]. Akurasi, Precision, Recall dan F-Measures didefinisikan dengan Persamaan (10).

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F\text{-}measure = \frac{2 * Precision * Recall}{Precision + Recall} \quad \text{... (10)}$$

4. Hasil dan Pembahasan

Pengujian hasil menggunakan metode Naive Bayes Gain Ratio dilakukan dengan membandingkan hasil percobaan Naive Bayes tanpa menggunakan Gain Ratio. Pada Percobaan awal, telah dilakukan pengujian menggunakan metode Naive Bayes dilakukan dengan membandingkan hasil percobaan Naive Bayes tanpa menggunakan Naive Bayes Gain Ratio. Perbandingan dilakukan terhadap Dataset CM 1 dan data JM 1 dengan masing-masing 498 data dan 10885 data. Hasil yang dibandingkan adalah akurasi data yang dihasilkan dengan menghitung selisih antara Naive Bayes Gain Ratio dan Naive Bayes biasa.

Dilakukan uji coba 1 pada data CM1 terhadap metode Naïve Bayes dengan menggunakan data latih menghasilkan 84.94% dan data uji menghasilkan 85.34%. Setelah itu dilakukan pemilihan fitur menggunakan Gain Ratio dan terpilih lima atribut terbaik. Hasil metode usulan Naïve Bayes Gain Ratio setelah dilakukan pemilihan fitur pada data CM 1 yaitu 87.55% untuk data latih dan 87.55% untuk data testing.

Tabel 2. Hasil Uji Coba

Metode	Akurasi %			
	Uji Coba 1		Uji Coba 2	
	Train	Test	Train	Test
Naive Bayes (NB)	84.94	85.34	80.41	80.42
Naive Bayes Gain Ratio (NB GR)	**87.55**	**87.55**	**80.51**	**80.57**

Pada uji coba 2, dilakukan uji coba pada data JM1 terhadap metode Naïve Bayes dengan menggunakan data latih menghasilkan 80.41% dan data uji menghasilkan 80.42%. Setelah itu dilakukan pemilihan fitur menggunakan Gain Ratio dan terpilih lima atribut terbaik. Hasil metode usulan Naïve Bayes Gain Ratio setelah dilakukan pemilihan fitur pada data CM 1 yaitu 80.51% untuk data latih dan 80.57% untuk data testing. Hasil akurasi tersebut dapat dilihat pada Tabel 2.Penghitungan Bobot Gain Ratio.

Dari hasil uji coba 1 didapatkan nilai akurasi Naïve Bayes sebesar 84,94% dan 85,34% sedangkan nilai akurasi untuk metode yang diusulkan atau Weighted Naïve Bayes sebesar 87% dan 87,55%. Hasil metode yang diusulkan lebih tinggi disebabkan oleh pemberian bobot dan pemilihan fitur pada probabilitas dari setiap fitur pada data terhadap kategori. Pemberian bobot pada probabilitas mengakibatkan jarak antar peluang satu data terhadap kategori semakin jauh. Hasil dari penelitian yang diusulkan sesuai dengan penelitian Hilden, Ferreira dan Hall yang berpendapat bahwa pembobotan atribut kelas dapat meningkatkan pengaruh prediksi [15][16][17].

Akan tetapi pada uji coba 2, akurasi pada metode yang diusulkan cenderung rendah dibandingkan dengan Naïve Bayes biasa. Hal ini dikarenakan data yang sering muncul pada seluruh kategori menghasilkan nilai Gain Ratio yang tinggi dan mengakibatkan terjadinya kesalahan klasifikasi. Setelah diketahui hasil akurasi pada uji coba 2 rendah. Maka, dilakukan proses pemilihan fitur terbaik untuk mengatasi kesalahan klasifikasi yang disebabkan oleh sering munculnya term pada seluruh dokumen. Dari hasil uji coba pemilihan fitur didapatkan akurasi sebesar 80,51% dan 80,57% untuk

metode usulan dan 80,41% dan 80,42% untuk metode Naïve Bayes biasa. Hal ini dikarenakan data yang sering muncul pada kelas lain terdapat pula pada kelas yang diuji. Hal ini dikarenakan data yang digunakan pada kelas yang diuji merepresentasikan kelas tersebut. Sehingga pada uji coba ini diketahui bahwa pemilihan fitur terbaik dapat mengurangi jumlah data yang sering muncul pada kelas lain.

5. Simpulan

Metode Naïve Bayes Gain Ratio dapat mengoptimalkan nilai akurasi metode Naïve Bayes biasa. Hal ini dapat dilihat dari hasil akurasi Naïve Bayes Gain Ratio sebesar 87,55% dan 80,57% dibandingkan dengan Naïve Bayes biasa sebesar 85,34% dan 80,42%. Metode usulan dapat menghasilkan tingkat akurasi yang lebih tinggi dikarenakan setiap probabilitas dari atribut diberi bobot yang menghasilkan nilai yang lebih tinggi. Ketika dilakukan pemilihan fitur terbaik didapatkan akurasi sebesar 87,55% untuk metode usulan dan 85,34% untuk metode Naïve Bayes biasa. Hal ini dapat disimpulkan bahwa pemilihan fitur mengguankan Gain Ratio pada Naive Bayes dapat mengatasi kesalahan klasifikasi pada Prediksi Cacat Perangkat Lunak.

Daftar Pustaka

[1] Arar, Ömer Faruk, Ayan, Kürsat. Software defect prediction using cost-sensitive neural network. Applied Soft Computing 33 (2015) 263–277.

[2] Turhan, Burak, Bener, Ayse.*Analysis of Naive Bayes' assumptions on software fault data:An empirical study.* Data & Knowledge Engineering 68 (2009) 278–290.

[3] Elish, Mahmoud O. *Predicting defect-prone software modules using supportvector machines.* Journal of Systems and Software. 2008.

[4] ZhangJiang.Li. Kong.*Two Feature Weighting Approaches for Naive Bayes Text Kelasifiers.* Knowledge-Based System. 2016.

[5] Catal, Cagatay. *Investigating the effect of dataset size, metrics sets, and featureselection techniques on software fault prediction problem.* Information Sciences. 2009.

[6] Chen, Jingnian. *A selective Bayes Classifier for classifying incomplete data based on gain ratio.* Knowledge-Based Systems 21 (2008) 530–534. 2008.

[7] Socrates, Adi Guna. *Optimasi Naive Bayes Dengan Pemilihan Fitur Dan Pembobotan Gain Ratio.* Lontar Komputer : Jurnal Ilmiah Teknologi Informasi Vol. 7. 2016.

[8] Hamzah, Amir. *Klasifikasi Teks Dengan Naïve Bayes Classifier (Nbc) Untuk Pengelompokan Teks Berita Dan Abstract Akademis.* Seminar Nasional Aplikasi Sains & Teknologi (SNAST) Periode III. 2012.

[9] Zaidi, Cerquides Carman. Alleviating Naive Bayes Attribute Independence Assumption by Attribute Weighting. Journal of Machine Learning Research 14 1947-1988. 2013.

[10] Pressman, R.S. *Software Engineering: A Practitioner's Approach*. McGraw-Hill, NY. 2001

[11] Boehm, B., Bassili, V. *Software defect top 10 list.* IEEE Computer 34(1):2—6. 2001

[12] Arar, Ömer Faruk, Ayan, Kürsat. Software defect prediction using cost-sensitive neural network. Applied Soft Computing 33 (2015) 263–277.

[13] Turhan, Burak, Bener, Ayse.*Analysis of Naive Bayes' assumptions on software fault data:An empirical study.* Data & Knowledge Engineering 68 (2009) 278–290.

[14] Elish, Mahmoud O. *Predicting defect-prone software modules using supportvector machines.* Journal of Systems and Software. 2008.

[15] Hilden dan B. Bjerregaard. *Computer-aided diagnosis and the atypical case*. In Decision Making and Medical Care: Can Information Science Help. North-Holland Publishing Company. 1976: 365–378.

[16] T. A. S. Ferreira, D. G. T. Denison, dan D. J. Hand. *Weighted naive Bayes modelling for data mining*. 2001

[17] A. Hall. *A decision tree-based attribute weighting filter for naive Bayes*. Knowledge-Based Systems, 2007; 20:120–126.