Integrasi Bagging dan Greedy Forward Selection pada Prediksi Cacat *Software* dengan Menggunakan Naïve Bayes

Fitriyani[1] dan Romi Satria Wahono[2]

[1]*Fakultas Ilmu Komputer, STMIK Nusa Mandiri*
fitriyani.fitriyani@hotmail.co.id

[2] *Fakultas Ilmu Komputer, Universitas Dian Nuswantoro*
romi@romisatriawahono.net

Abstrak: Kualitas *software* ditemukan pada saat pemeriksaan dan pengujian. Apabila dalam pemeriksan atau pengujian tersebut terdapat cacat *software* maka hal tersebut akan membutuhkan waktu dan biaya dalam perbaikannya karena biaya untuk estimasi dalam memperbaiki *software* yang cacat dibutuhkan biaya yang mencapai 60 Miliar pertahun. Naïve bayes merupakan algoritma klasifikasi yang sederhana, mempunya kinerja yang bagus dan mudah dalam penerapannya, sudah banyak penelitian yang menggunakan algoritma naïve bayes untuk prediksi cacat *software* yaitu menentukan *software* mana yang masuk kategori cacat dan tidak cacat pada. Dataset NASA MDP merupakan dataset publik dan sudah banyak digunakan dalam penelitian karena sebanyak 64.79% menggunakan dataset tersebut dalam penelitian prediksi cacat *software*. Dataset NASA MDP memiliki kelemahan adalah kelas yang tidak seimbang dikarenakan kelas mayoritas berisi tidak cacat dan minoritas berisi cacat dan kelemahan lainnya adalah data tersebut memiliki dimensi yang tinggi atau fitur-fitur yang tidak relevan sehingga dapat menurunkan kinerja dari model prediksi cacat *software*. Untuk menangani ketidakseimbangan kelas dalam dataset NASA MDP adalah dengan menggunakan metode ensemble (bagging), bagging merupakan salah satu metode ensemble untuk memperbaiki ketidakseimbangan kelas. Sedangkan untuk menangani data yang berdimensi tinggi atau fitur-fitur yang tidak memiliki kontribusi dengan menggunakan seleksi fitur greedy forward selection. Hasil dalam penelitian ini didapatkan nilai AUC tertinggi adalah menggunakan model naïve bayes tanpa seleksi fitur adalah 0.713, naïve bayes dengan greedy forward selection sebesar 0.941 dan naïve bayes dengan greedy forward selection dan bagging adalah sebesar 0.923. Akan tetapi, dilihat dari rata-rata peringkat bahwa naïve bayes dengan greedy forward selection dan bagging merupakan model yang terbaik dalam prediksi cacat *software* dengan rata-rata peringkat sebesar 2.550.

Kata Kunci: Naïve Bayes, Greedy Forward Selection, Bagging, Ketidakseimbangan Kelas, Fitur-Fitur yang tidak Relevan.

1 PENDAHULUAN

Prediksi cacat *software* merupakan salah satu kegiatan penting pada fase atau tahap *testing* dalam *Software Development Life Cycle* (Arora, Tetarwal, & Saha, 2015). Terbukti pada tahun 2002, menurut NIST (*National Institute of Standards and Technology*) estimasi biaya cacat *software* mencapai $60 billion atau sekitar 60 miliar pertahun (Strate & Laplante, 2013) dan lebih dari 30 tahun prediksi cacat *software* merupakan topik yang penting dalam *software engineering*. Prediksi yang akurat pada kesalahan yang mungkin dapat terjadi dalam kode merupakan hal yang penting karena dapat membantu pada saat tahap pengujian, pengurangan biaya dan peningkatan kualitas perangkat lunak (Song, Jia, Shepperd, Ying, & Liu, 2010).

Kualitas dianggap sebagai isu penting dalam bidang *software engineering*, akan tetapi membangun kualitas perangkat lunak sangat membutuhkan biaya yang mahal untuk meningkatkan efektifitas dan efisiensi jaminan kualitas dan pengujian (Chang, Mu, & Zhang, 2011). Prediksi cacat *software* ini berupaya untuk meningkatkan kualitas perangkat lunak dan efisiensi pengujian dengan membangun model prediksi klasifikasi dari atribut kode untuk mengidentifikasi secara tepat pada modul rawan kesalahan (Lessmann, Baesens, Mues, & Pietsch, 2008). Saat ini penelitian tentang prediksi cacat *software* berfokus pada metode klasifikasi sebanyak 77.46%, 14.08% menggunakan metode estimasi dan sebanyak 1.41% menggunakan metode *clustering* dan asosiasi (Wahono, 2015).

Dataset NASA MDP merupakan data metrik perangkat lunak yang kerap kali digunakan dalam penelitian *software defect prediction* atau prediksi cacat *software*. Dataset NASA mudah diperoleh dan tersedia untuk umum karena sebanyak 64.79% penelitian menggunakan publik dataset dan 35.21% penelitian menggunakan dataset privat (Wahono, 2015).

Masalah dalam *software defect prediction* adalah *redundant data*, korelasi, fitur yang tidak relevan, *missing samples* dan masalah ini dapat membuat dataset tidak seimbang karena sulit untuk memastikan antara data cacat atau tidak cacat (Laradji, Alshayeb, & Ghouti, 2015). *Feature selection* (seleksi fitur atau atribut) dapat menangani untuk mengurangi masalah *redudant data* dan fitur yang tidak relevan. Seleksi fitur merupakan langkah yang penting dalam mesin pembelajaran (*machine learning*) (Laradji, Alshayeb, & Ghouti, 2015). Salah satu metode seleksi fitur adalah menggunakan greedy forward selection. Seleksi fitur dengan greedy forward selection sangat efisien, sederhana dan tidak seperti teknik seleksi fitur yang membutuhkan waktu lama dalam prosesnya (forward selection dan backward elimination). Greedy forward selection hanya memilih fitur yang berkontribusi dan dapat meningkatkan performa klasifikasi (Laradji, Alshayeb, & Ghouti, 2015).

Selain masalah *redudant data* dan fitur-fitur yang tidak relevan, pada dataset cacat *software* ditemukan dataset yang

tidak seimbang (*imbalance class*) karena data yang cacat jumlahnya lebih sedikit dibandingkan dengan data yang tidak cacat, sehingga data yang termasuk kelas mayoritas adalah tidak cacat dan data yang termasuk kelas minoritas adalah cacat. Untuk menangani masalah dalam *imbalance class* salah satunya adalah menggunakan teknik ensemble (boosting dan bagging). Bagging (bootstrap aggregating) merupakan teknik yang dapat meningkatkan klasifikasi dengan kombinasi klasifikasi secara acak pada dataset *training* dan bagging juga dapat mengurangi variansi dan menghindari *overfitting* (Wahono & Suryana, Combining Particle Swarm Optimization based Feature Selection and Bagging Technique for Software Defect, 2013).

Banyak penelitian yang telah dilakukan sebelumnya dan algoritma naive bayes merupakan model yang terbaik dibandingkan model lainnya seperti: logistic regression, neural network, random forest, decission tree, support vector machine dan k-nearest neigbor. Naive bayes merupakan algoritma klasifikasi yang sederhana dan mudah diimplementasikan sehingga algoritma ini sangat efektif apabila diuji dengan dataset yang tepat, terutama bila naïve bayes dengan seleksi fitur, maka naive bayes dapat mengurangi *redudant* pada data (Witten, Frank, & Hall, 2011). Algoritma naive bayes termasuk dalam *supervised learning* dan salah satu algoritma pembelajaran tercepat yang dapat menangani sejumlah fitur atau kelas (Lee, 2015).

Pada penelitian ini untuk menangani masalah *redundant data* menggunakan seleksi fitur greedy forward selection dan untuk menangani *imbalance class* menggunakan teknik ensemble bagging, sedangkan algoritma klasifikasi yang digunakan dalam penelitian ini adalah algoritma naive bayes. Pada penelitian ini menggunakan dataset NASA (*National Aeronautics and Space Administration*) MDP (*Metrics Data Program*) *repository* sebagai *software metrics*.

2 PENELITIAN TERKAIT

Banyak penelitian tentang prediksi cacat *software* dan sudah banyak metode yang dalam penelitian tersebut menghasilkan kinerja yang baik, akan tetapi sebelum dilakukan penelitian adalah mengkaji metode-metode yang sudah menggunakan metode yang lain dalam penelitian sebelumnya, sehingga dapat mengetahui *state of the art* dalam penelitian yang membahas tentang fitur-fitur yang tidak relevan (*irrelevant features*) dan ketidakseimbangan kelas (*imbalanced class*).

Penelitian yang dilakukan oleh Song, Jia, Shepperd, Ying dan Liu bahwa prediksi cacat *software* merupakan topik penelitian yang penting dan lebih dari 30 tahun penelitian prediksi cacat *software* memfokuskan pada estimasi jumlah cacat pada sistem *software*, menemukan asosiasi atau hubungan antara cacat, mengklasifikasikan antara cacat dan tidak cacat (Song, Jia, Shepperd, Ying, & Liu, 2010). Pada penelitian ini, *data preprocessing* merupakan bagian yang penting dalam menangani *missing values*, *discretizing* atau transformasi untuk atribut *numeric* dan menangani masalah tersebut dapat menggunakan metode *log filtering*. Setelah proses *data prosessing* pada prediksi cacat *software*, dilakukan pemilihan atribut terbaik atau dapat disebut dengan *attribute selection*. Dalam penelitiani ini untuk pemilihan atribut terbaik menggunakan *forward selection* dan *backward elimination* dan algoritma pembelajarannya yang digunakan adalah naïve bayes, J48 dan OneR. Hasilnya dapat diketahui bahwa naïve bayes dengan *log filtering* dan seleksi fitur (*forward selection* dan *backward elimination*) lebih baik daripada algoritma J48 dan OneR untuk meningkatkan AUC.

Pada penelitian (Khoshgoftaar, Hulse, & Napolitano, 2011) *noisy* dan *imbalance class* lebih efektif ditangani dengan teknik bagging dan boosting karena *imbalance class* dan *noise* dapat berpengaruh pada kualitas data dalam hal kinerja klasifikasi. Dataset dalam penelitian ini menggunakan Letter A, Nursey3, OptDigits8, Splice2, sedangkan metode dalam penelitian ini adalah menggunakan teknik data *sampling* seperti *Random Undersampling* (RUS) dan *Synthetic Minority Oversampling Technique* (SMOTE) yang menggabungkan dengan teknik bagging dan boosting sehingga penelitian ini menggunakan metode *Exactly Balanced Bagging* (EBBag), *Roughly Balanced Bagging* (RBBag), SMOTE dan Boosting (SMOTEBoost), RUS dan Boosting (RUSBoost), EBBag dan RBBag dengan *Replacement* (EBBagR dan RBBagR), EBBag dan RBBag tanpa *Replacement* (EBBagN dan RBBagN), SMOTE dan Boosting dengan *Reweighting* (SMOTEBoostW), RUS dan Boosting dengan *Reweigthting* (RUSBoostW), SMOTE dan Boosting dengan *Resampling* (SMOTEBoostS), RUS dan Boosting dengan *Resampling* (RUSBoostS). Algoritma pembelajaran menggunakan C4.5D (*Default Parameter*) dan C4.5N (*disable* pruning dan *enable laplace smoothing*), naïve bayes dan RIPPER. Hasil akhir dari penelitian ini adalah bahwa teknik bagging lebih baik tanpa menggunakan *replacement* untuk pembelajaran pada *noisy* dan *imbalance class*, walaupun teknik boosting lebih popular untuk menangani *imbalance class*, sehingga metode bagging lebih baik tanpa *replacement* untuk menangani *noisy* dan *imbalance class* dengan AUC tertinggi sebesar 0.947.

Penelitian yang dilakukan oleh Ma, Luo, Zeng dan Chen bahwa prediksi kualitas modul *software* merupakan hal yang sangat kritis, dalam penelitian ini (Ma, Luo, Zeng, & Chen, 2012) menggunakan algoritma klasifikasi naïve bayes di weka dengan sebutan metode CC, *nearest neighbor filter* dengan sebutan NN-*filter* dan transfer naïve bayes dengan sebutan TNB. Dataset dalam penelitian ini menggunakan dataset NASA: KC1, MC2, KC3, MW1, KC2, PC1, CM1. Hasilnya bahwa menggunakan algoritma TNB dapat meningkatkan kinerja menjadi lebih bagus dan dapat meningkatkan AUC sebesar 0.6236 pada KC1, 0.6252 pada MC2, 0.7377 pada KC3, 0.6777 pada MW1, 0.7787 pada KC2, 0.5796 pada PC1 dan 0.6594 pada CM1. AUC lebih meningkat dengan menggunakan algoritma TNB dan AUC dengan hasil AUC tertinggi sebesar 0.77 pada dataset KC2.

Pada penelitian (Wahono & Herman 2014), metaheuristic optimization untuk *feature selection* dapat menggunakan Genetic Algorithm (GA) dan Particle Swarm Optimization (PSO) dan untuk dapat lebih meningkatkan akurasi dari prediksi cacat *software* menggunakan metode bagging. Algoritma klasifikasi pada penelitian ini sebanyak 10 algoritma seperti: (Logistic Regression (LR), Linear Discriminant Analysis (LDA), dan Naïve Bayes (NB)), Nearest Neighbors (k-Nearest Neighbor (k-NN) dan K*), Neural Network (Back Propagation (BP)), Support Vector Machine (SVM), dan Decision Tree (C4.5, *Classification* and *Regression Tree* (CART), dan Random Forest (RF)). Dataset dalam penelitian ini menggunakan dataset NASA MDP dengan menggunakan 9 dataset sebagai berikut: CM1, KC1, KC3, MC2, MW1, PC1, PC2, PC3, PC4. Hasil akhir dari metode ini dapat menghasilkan peningkatan yang mengesankan pada banyak metode klasifikasi dan berdasarkan hasil perbandingan antara GA dan PSO tidak ada perbedaan yang signifikan ketika menggunakan *feature selection* pada banyak metode klasifikasi dalam prediksi cacat *software*. AUC rata-rata meningkat 25.99% untuk GA dan PSO meningkat 20.41%.

3 METODE YANG DIUSULKAN

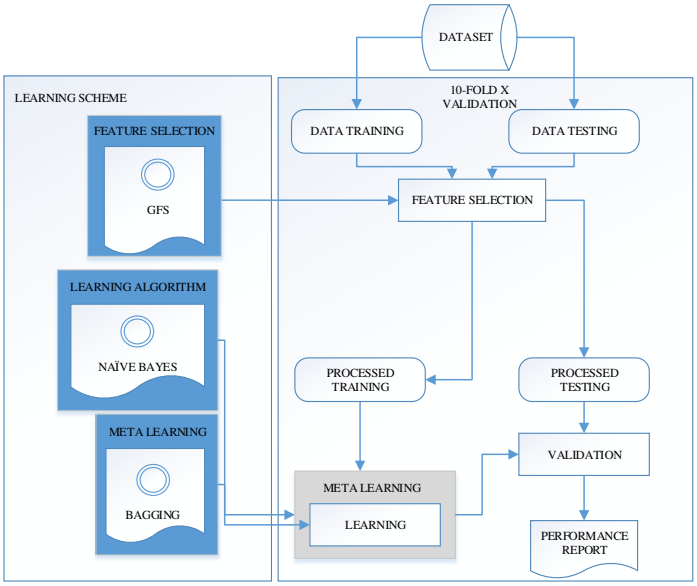
Penelitian ini dilakukan dengan mengusulkan model, melakukan eksperimen dengan menguji model yang diusulkan, evaluasi dan validasi, mengembangkan aplikasi. Pada penelitian ini dataset yang digunakan adalah dataset NASA (*National Aeronautics and Space Administration*) MDP (*Metrics Data Program*), dataset yang digunakan sebanyak 10 dataset yaitu: CM1, JM1, KC1, KC3, MC2, PC1, PC2, PC3, PC4 DAN PC5. Tabel 1 merupakan spesifikasi dari dataset NASA MDP.

Tabel 1 Spesifikasi Dataset NASA MDP

Atribut	NASA MDP									
	CM1	JM1	KC1	KC3	MC2	PC1	PC2	PC3	PC4	PC5
LOC_BLANK	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
LOC_CODE_AND_COMMENT	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
LOC_COMMENTS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
LOC_EXECUTABLE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
LOC_TOTAL	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NUMBER_OF_LINES	✓			✓	✓	✓	✓	✓	✓	✓
HALSTEAD_CONTENT	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
HALSTEAD_DIFFICULTY	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
HALSTEAD_EFFORT	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
HALSTEAD_ERROR_EST	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
HALSTEAD_LENGTH	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
HALSTEAD_LEVEL	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
HALSTEAD_PROG_TIME	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
HALSTEAD_VOLUME	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NUM_OPERANDS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NUM_OPERATORS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NUM_UNIQUE_OPERANDS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NUM_UNIQUE_OPERATORS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CYCLOMATIC_COMPLEXITY	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CYCLOMATIC_DENSITY	✓			✓	✓	✓	✓	✓	✓	✓
DESIGN_COMPLEXITY	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ESSENTIAL_COMPLEXITY	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BRANCH_COUNT	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CALL_PAIRS	✓			✓	✓	✓	✓	✓	✓	✓
CONDITION_COUNT	✓			✓	✓	✓	✓	✓	✓	✓
DECISION_COUNT	✓			✓	✓	✓	✓	✓	✓	✓
DECISION_DENSITY	✓			✓	✓	✓	✓	✓	✓	✓
DESIGN_DENSITY	✓			✓	✓	✓	✓	✓	✓	✓
EDGE_COUNT	✓			✓	✓	✓	✓	✓	✓	✓
ESSENTIAL_DENSITY	✓			✓	✓	✓	✓	✓	✓	✓
GLOBAL_DATA_COMPLEXITY				✓	✓					
GLOBAL_DATA_DENSITY				✓	✓					
MAINTENANCE_SEVERITY	✓			✓	✓	✓	✓	✓	✓	✓
MODIFIED_CONDITION_COUNT	✓			✓	✓	✓	✓	✓	✓	✓
MULTIPLE_CONDITION_COUNT	✓			✓	✓	✓	✓	✓	✓	✓
NODE_COUNT	✓			✓	✓	✓	✓	✓	✓	✓
NORMALIZED_CYLOMATIC	✓			✓	✓	✓	✓	✓	✓	✓
PARAMETER_COUNT	✓			✓	✓	✓	✓	✓	✓	✓
PERCENT_COMMENTS	✓			✓	✓	✓	✓	✓	✓	✓
PATHOLOGICAL_COMPLEXITY										
Defective	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Jumlah atribut	**37**	**21**	**21**	**39**	**39**	**37**	**37**	**37**	**37**	**37**
Jumlah modul	**344**	**9591**	**2095**	**200**	**127**	**759**	**1586**	**1125**	**1399**	**17001**
Jumlah modul cacat	**42**	**1759**	**325**	**36**	**44**	**61**	**16**	**140**	**178**	**503**
Presentase modul cacat	**12%**	**18%**	**16%**	**18%**	**35%**	**8%**	**1%**	**12%**	**13%**	**3%**
Jumlah modul tidak cacat	**302**	**7834**	**1771**	**164**	**83**	**698**	**1569**	**985**	**1221**	**16498**

Pada penelitian ini mengusulkan model Naïve Bayes dan seleksi fitur Greedy Forward Selection (NB+GFS), model yang diusulkan selanjutnya adalah model Naïve Bayes dengan Greedy Forward Selection dan Bagging (NB+GFS+BG), model bagging untuk menangani permasalahan ketidakseimbangan kelas (*imbalance class*). Gambar 1 merupakan kerangka penelitian yang diusulkan.

Eksperimen dilakukan dengan menguji model menggunakan aplikasi Rapidminer, hasil kinerja dari model di evaluasi menggunakan friedman test untuk dapat membandingkan model-model yang diusulkan sehingga dapat diketahui model yang terbaik pada penelitian prediksi cacat *software*. Validasi yang digunakan dalam penelitian ini menggunakan 10-*fold cross validation*. Pengembangan aplikasi yang dilakukan menggunakan IDE (*Integrated Development Environment*) Netbeans dengan bahasa Java. Gambar 1 merupakan kerangka penelitian yang diusulkan.

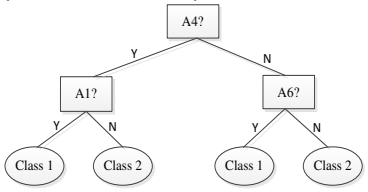


Gambar 1 Kerangka Penelitian

Pada model yang diusulkan dataset akan dibagi menjadi 10 bagian menggunakan 10-*fold cross validation*, data bagian pertama menjadi data testing dan data bagian kedua sampai dengan data bagian kesepuluh menjadi data training. Selanjutnya dilakukan seleksi fitur dengan menggunakan Greedy Forward Selection (GFS), sehingga dihasilkan fitur-fitur relevan yang dapat meningkatkan kinerja dari model prediksi cacat *software*. Data training yang sudah dibentuk dengan cross validation akan dibuat data training baru secara acak oleh model bagging berbasis naïve bayes sebanyak iterasi yang dimasukkan. Jika lebih banyak masuk dalam kategori cacat maka record tersebut di prediksi cacat, sebaliknya jika lebih banyak masuk dalam kategori tidak cacat, maka record tersebut di prediksi tidak cacat.

A. Seleksi Fitur Greedy Forward Selection

Subset selection merupakan metode *feature selection* (seleksi fitur), *subset selection* adalah menemukan *subset* terbaik (fitur), *subset* yang terbaik mempunyai jumlah dimensi yang paling berkontribusi pada akurasi. Seleksi fitur Seleksi fitur untuk menentukan atribut terbaik dan terburuk menggunakan information gain (Han, Kamber, & Pei, 2012). Seperti pada Gambar 2.

Forward Selection	Backward Elimination	Decision Tree *Induction*
Atribut asli: {A1, A2, A3, A4, A5, A6} Pengurangan atribut asli: {} =>{A1} =>{A1, A4} =>Pengurangan atribut: {A1, A4, A6}	Atribut asli: {A1, A2, A3, A4, A5, A6} =>{A1, A3, A4, A5, A6} =>{A1, A4, A5, A6} =>Pengurangan Atribut: {A1, A4, A6}	Atribut asli: {A1, A2, A3, A4, A5, A6}  =>Pengurangan Atribut: {A1, A3, A6}

Gambar 2 Seleksi atribut Greedy
Sumber: (Han, Kamber, & Pei, 2012)

Algoritma greedy dengan seleksi subset atribut (Han, Kamber, & Pei, 2012) sebagai berikut:

1. Stepwise forward selection

   Prosedur dimulai dengan himpunan kosong dari atribut sebagai set yang dikurangi, atribut yang terbaik

dari atribut asli ditentukan dan ditambahkan pada set yang kurang. Pada setiap iterasi berikutnya yang terbaik dari atribut asli yang tersisa ditambahkan ke set.

2. Stepwise backward elimination

Prosedur dimulai dengan *full* set atribut. Pada setiap langkah, teknik ini dapat menghilangkan atribut terburuk yang tersisa di dataset.

Atribut terbaik dan terburuk dapat ditentukan dengan menggunakan tes signifikansi statistik yang berasumsi bahwa atribut tidak saling berhubungan dengan satu sama lain (independen) (Han, Kamber, & Pei, 2012). Untuk langkah-langkah dalam menentukan evaluasi dalam pemilihan atribut dapat menggunakan information gain yang digunakan dalam pohon keputusan *decision tree* (Han, Kamber, & Pei, 2012).

Seleksi atribut menggunakan information gain adalah memilih gain tertinggi dan formula yang digunakan adalah sebagai berikut dimana langkah yang pertama adalah dengan mencari nilai entropy sebagai berikut:

$$Info(D) = -\sum_{i=1}^{m} p_i log_2(p_i)$$

$Info(D)$ adalah untuk mengetahui nilai dari entropy, kemudian jika suatu atribut mempunyai nilai yang berbeda-beda maka menggunakan formula sebagai berikut:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} x\ Info(D_j)$$

Kemudian untuk mendapatkan hasil *gain*, dimana formulanya menjadi:

$$Gain(A) = Info(D) - Info_A(D)$$

B. Bagging

Teknik ensemble merupakan teknik yang sukses untuk menangani dataset yang tidak seimbang meskipun tidak secara khusus dirancang untuk masalah data yang tidak seimbang (Laradji, Alshayeb, & Ghouti, 2015). Teknik bagging merupakan salah satu teknik ensemble dan teknik ini pada klasifikasi memisahkan data *training* ke dalam beberapa data *training* baru dengan *random sampling* dan membangun model berbasis data *training* baru (Wahono & Suryana, 2013). Algoritma bagging untuk klasifikasi (Liu & Zhou, 2013):

Input: Data set D=$\{(x_i, y_i)\}_{i=1}^{n}$
Base learning algorithm $\mathfrak{L}$
The number of iterations T
1. for t=1 to T do
2. $h_t = \mathfrak{L}(D, \mathcal{D}_{bs})$ /* $\mathcal{D}_{bs}$ *is the bootstrap distribution* */
3. end for

Output: $H(x)=\max_y \sum_{t=1}^{n} I(h_t(x) = y)$ /* I(x)=1 if x is true, and 0 otherwise */

C. Naïve bayes

Klasifikasi menggunakan naïve bayes dapat menghasilkan akurasi yang tinggi dan cepat ketika diaplikasikan pada data yang besar (Han, Kamber, & Pei, 2012). Pengklasifikasi naïve bayes berpendapat bahwa nilai dari atribut pada kelas tertentu tidak bergantung (*independence*) pada nilai atribut lainnya, pendapat ini dapat disebut *class-conditional independence* sehingga perhitungannya dapat dibuat lebih sederhana dan disebut "naif (naïve)" (Han, Kamber, & Pei, 2012).

Persamaan naïve bayes dengan menggunakan distribusi gaussian karena dataset NASA MDP merupakan data tipe numerik. Pada distribusi gaussian, dihitung mean μ dan standar deviasi σ pada semua atribut, berikut adalah persamaan yang digunakan:

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Pengukuran kinerja model menggunakan confusion matrix, *confusion matrix* merupakan alat untuk menganalisa seberapa baik kinerja dari pengklasifikasi dapat mengenali tupel dari kelas yang berbeda (Han, Kamber, & Pei, 2012). Confusion matrix memberikan penilaian kinerja klasifikasi berdasarkan objek dengan benar atau salah (Gorunescu, 2011). Confusion matrix merupakan matrik 2 dimensi yang menggambarkan perbandingan antara hasil prediksi dengan kenyataan.

Berikut adalah persamaan model confusion matrix:

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

$$Sensitivitas = recall = TP_{rate} = \frac{TP}{TP\ +\ FN}$$

$$Specificity = TN_{rate} = \frac{TN}{TN\ +\ FP}$$

$$FP_{rate} = \frac{FP}{FP\ +\ TN}$$

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

$$F - Measure = \frac{(1 + \beta^2)\ x\ recall\ x\ precision}{(\beta\ x\ recall\ +\ precision)}$$

$$G - Mean = \sqrt{Sensitivitas\ x\ Specificity}$$

F-measure mengkombinasikan *recall* atau *sensitivity* dan *precision* sehingga menghasilkan metrik yang efektif untuk pencarian kembali informasi dalam himpunan yang mengandung masalah ketidakseimbangan.

Pada kelas yang tidak seimbang karena kelas minoritas mendominasi sehingga pengukuran kinerja yang tepat menggunakan Area *Under the ROC (Receiver Operating Characteristic) Curve (AUC)*, *f-measure*, *Geometric Mean (G-Mean)*. *Area Under ROC Curve (AUC)* digunakan untuk memberikan metrik numerik single untuk dapat membandingkan kinerja dari model, nilai AUC berkisar dari 0 sampai 1 dan model yang lebih baik prediksinya adalah yang mendekati nilai 1 (Gao, Khoshgoftaar, & Wald , 2014). Berikut adalah persamaan model AUC:

$$\text{AUC} = \frac{1+TP_{rate}-FP_{rate}}{2}$$

Pedoman umum yang digunakan untuk klasifikasi akurasi sebagai berikut:

1. 0.90-1.00 = *excellent classification*
2. 0.80-0.90 = *good classification*
3. 0.70-0.80 = *fair classification*
4. 0.60-0.70 = *poor classification*
5. 0.50-0.60 = *failure*

Kurva ROC adalah grafik antara sensitivitas (true *positive* rate) pada sumbu Y dengan 1- spesifisitas pada sumbu X (false *positive* rate), curve ROC ini seakan-akan menggambarkan tawar-menawar antara sumbu Y atau sensitivitas dengan sumbu X atau spesifisitas. Kurva ROC biasanya digunakan dalam pembelajaran dan data mining, nilai dalam kurva ROC dapat menjadi evaluasi sehingga dapat membandingkan algoritma. Dalam klasifikasi kurva ROC merupakan teknik untuk memvisualisasikan, mengatur dan memilih klasifikasi berdasarkan kinerja dari algoritma (Gorunescu, 2011).

Setelah dilakukan evaluasi terhadap model yang diusulkan, tahap selanjutnya adalah menganalisis model yang diusulkan dengan menggunakan metode statistik uji t (*t-test*) dan uji friedman (*friedman test*).

Uji t sampel berpasangan (paired-samples t-test) merupakan prosedur yang digunakan untuk membandingkan rata-rata dari dua variabel, variabel sebelum dan sesudah menggunakan model. Uji t dapat disebut juga dengan z-test dalam statistik dan pengujian ini dilakukan untuk mengetahui apakah ada perbedaan yang signifikan pada dua variabel yang diuji.

Algoritma untuk perbandingan performa z-test (Gorunescu, 2011) sebagai berikut:

Input:

- Masukkan bagian P_1 (akurasi klasifikasi) untuk sampel pertama (model M_1);
- Masukkan bagian P_2 (akurasi klasifikasi) untuk sampel kedua (model M_2);
- Masukkan ukuran sampel pada N_1 untuk sampel pertama;
- Masukkan ukuran sampel pada N_2 untuk sampel kedua;

  Kemudian $p - level$ dihitung berdasarkan nilai t untuk perbandingan dengan persamaan berikut ini:

$$|t| = \sqrt{\frac{N_1.N_2}{N_1.N_2}} . \frac{|P_1 - P_2|}{\sqrt{p.q}}$$

Dimana:

$$p = \frac{P_1.N_1 + P_2.N_2}{N_1 + N_2}, q = 1 - p, dan\ untuk\ N_1 + N_2 - 2$$

Output:

Level signifikansi untuk akurasi yang berbeda dari dua model.

Dalam statistik jika diketahui p > 0.05 berarti tidak ada perbedaan yang signifikan, akan tetapi jika diketahui nilai p < 0.05 maka ada perbedaan yang signifikan antara dua model.

4 HASIL EKSPERIMEN

Eksperimen pada penelitian ini menggunakan laptop LENOVO G470 dengan prosesor Intel Celeron CPU B800 @ 1.50 GHz, memori (RAM) 2.00 GB dan sistem operasi Windows 8.1 Pro 32-bit. Aplikasi yang dikembangkan menggunakan IDE (Integrated Development Environment) NetBeans menggunakan bahasa Java. Aplikasi yang digunakan dalam penelitian ini adalah Rapidminer, hasil kinerja dari model selanjutnya di analisis menggunakan menggunakan Microsoft Excel dan XLSTAT.

Hasil pengukuran pada penelitian ini dapat dilihat pada Tabel 1, dimana Tabel 1 adalah hasil akurasi dari model Naïve Bayes (NB), model Naïve Bayes dan Greedy Forward Selection (NB+GFS), model Naïve Bayes dengan Greedy Forward Selection dan Bagging (NB+GFS+BG). Tabel 2 merupakan hasil pengukuran sensitifitas model NB, NB+GFS dan NB+GFS+BG. Tabel 3 menunjukkan hasil pengukuran f-measure model NB, NB+GFS, NB+GFS+BG. Hasil penguruan g-mean dapat dilihat pada Tabel 4 dan Tabel 5 merupakan hasil AUC pada model NB, model NB+GFS dan model NB+GFS+BG.

Tabel 1 Hasil Akurasi

Model	CM1	JM1	KC1	KC3	MC2	PC1	PC2	PC3	PC4	PC5
NB	82.56%	81.31%	82.25%	79.00%	72.37%	88.54%	95.59%	33.53%	87.42%	96.63%
NB+GFS	86.94%	81.48%	83.64%	82.00%	75.64%	91.84%	96.66%	84.00%	83.99%	97.21%
NB+GFS+BG	85.76%	81.58%	83.35%	83.50%	74.87%	91.18%	96.91%	83.56%	86.77%	97.35%

Tabel 2 Hasil Sensitifitas

Model	CM1	JM1	KC1	KC3	MC2	PC1	PC2	PC3	PC4	PC5
NB	33.33%	94.86%	37.54%	36.11%	35.83%	36.07%	18.75%	90.00%	94.10%	44.73%
NB+GFS	34.67%	95.25%	33.54%	33.33%	43.18%	36.07%	6.25%	37.86%	87.39%	33.80%
NB+GFS+BG	28.57%	95.97%	33.85%	41.67%	50%	55.56%	12.50%	36.43%	91.07%	29.62%

Tabel 3 Hasil F-measure

Model	CM1	JM1	KC1	KC3	MC2	PC1	PC2	PC3	PC4	PC5
NB	0.55	0.95	0.58	0.57	0.57	0.58	0.43	0.48	0.94	0.66
NB+GFS	0.39	0.89	0.39	0.40	0.55	0.42	0.04	0.37	0.91	0.42
NB+GFS+BG	0.33	0.90	0.39	0.48	0.58	0.45	0.20	0.36	0.92	0.40

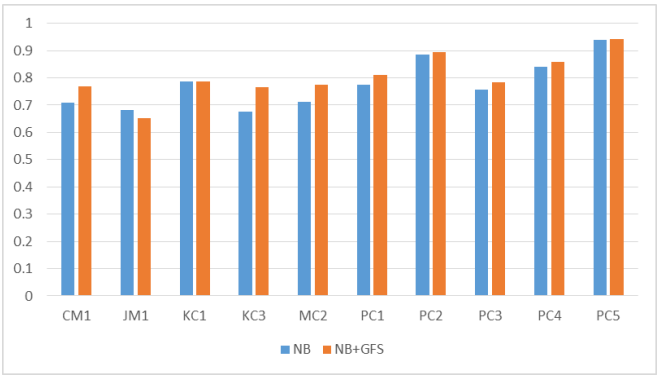
Tabel 4 Hasil G-mean

Model	CM1	JM1	KC1	KC3	MC2	PC1	PC2	PC3	PC4	PC5
NB	0.546	0.461	0.578	0.567	0.551	0.579	0.425	0.472	0.625	0.66
NB+GFS	0.57	0.44	0.56	0.56	0.63	0.59	0.25	0.59	0.73	0.58
NB+GFS+BG	0.52	0.42	0.56	0.62	0.66	0.71	0.35	0.57	0.72	0.54

Tabel 5 Hasil AUC

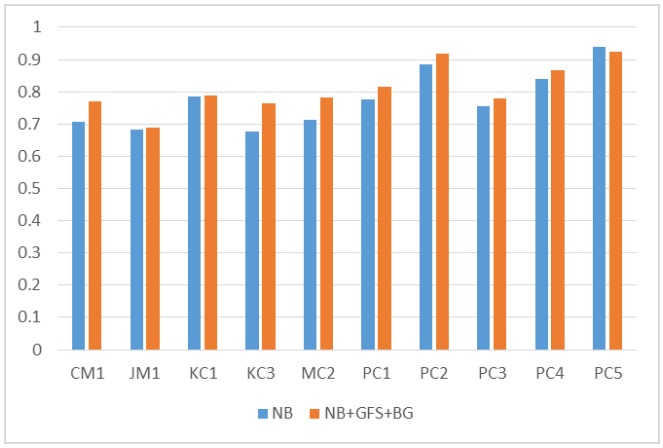
Model	CM1	JM1	KC1	KC3	MC2	PC1	PC2	PC3	PC4	PC5
NB	0.708	0.683	0.786	0.677	0.712	0.775	0.885	0.756	0.84	0.94
NB+GFS	0.769	0.653	0.788	0.765	0.775	0.81	0.894	0.784	0.86	0.941
NB+GFS+BG	0.771	0.688	0.788	0.763	0.781	0.817	0.918	0.778	0.866	0.923

Gambar 3 merupakan grafik perbandingan AUC model Naïve Bayes (NB) dengan model Naïve Bayes dan Greedy Forward Selection (NB+GFS).



Gambar 3 Grafik AUC Model NB dan NB+GFS

Gambar 4 menunjukkan grafik perbandingan AUC antara model Naïve Bayes (NB) dengan model Naïve Bayes dan Greedy Forward Selection dengan Bagging (NB+GFS+BG).



Gambar 4 Perbandingan AUC Model NB dan NB+GFS+BG

Hasil dari pengukuran kinerja, selanjutnya di analisis menggunakan uji t (*t-test*) untuk mengetahui model yang terbaik. Uji t dilakukan dengan membandingkan dua model dan mengukur p-value, jika p-value < nilai alpha (0.05), maka ada perbedaan yang signifikan antara dua model yang dibandingkan. Sebaliknya, jika p-value > nilai alpha, maka tidak ada perbedaan yang signifikan.

Uji t dilakukan pada AUC dengan menggunakan metode statistik untuk menguji hipotesa pada Naïve Bayes (NB) dengan Naïve Bayes dan Greedy forward selection (NB+GFS).

H_0: Tidak ada perbedaan nilai rata-rata AUC NB dengan NB+GFS.

H_1: Ada perbedaan antara nilai rata-rata AUC NB dengan NB+GFS.

Tabel 6 Uji t Model NB dan Model NB+GFS

	NB	*NB+GFS*
Mean	0.7762	**0.8039**
Variance	0.007838178	0.006342767
Observations	10	10
Pearson Correlation	0.917706055	
Hypothesized Mean Difference	0	
df	9	
t Stat	-2.487968197	
P(T<=t) one-tail	0.017268476	
t Critical one-tail	1.833112933	
P(T<=t) two-tail	**0.034536952**	
t Critical two-tail	2.262157163	

Pada Tabel 6 dapat diketahui bahwa nilai rata-rata AUC dari model NB+GFS lebih tinggi dibandingkan model NB sebesar 0.8039. Dalam uji beda statistik nilai alpha ditentukan sebesar 0.05, jika nilai p lebih kecil dibanding alpha ($p < 0.05$) maka H_0 ditolak dan H_1 diterima sehingga ada perbedaan yang signifikan antara model yang dibandingkan, akan tetapi jika nilai p lebih besar dibandingkan nilai alpha ($p > 0.05$) maka H_0 diterima dan H_1 ditolak sehingga tidak ada perbedaan yang signifikan antara model yang dibandingkan.

Pada Tabel 6 dapat dilihat bahwa nilai P(T<=t) adalah 0.03 dan ini menunjukkan bahwa nilai p lebih kecil dibandingkan nilai alpha (0.03 < 0.05) sehingga hipotesis H_0 ditolak dan H_1 diterima. Hipotesis H_1 diterima berarti ada perbedaan signifikan antara model NB dan model NB+GFS sehingga model NB+GFS membuat peningkatan ketika dibandingkan dibandingkan dengan modek NB.

Tabel 7 Uji t Model NB dan Model NB+GFS+BG

	NB	*NB+GFS+BG*
Mean	0.7762	**0.8093**
Variance	0.007838178	0.005397344
Observations	10	10
Pearson Correlation	0.936299769	
Hypothesized Mean Difference	0	
df	9	
t Stat	-3.221564827	
P(T<=t) one-tail	0.005231557	
t Critical one-tail	1.833112933	
P(T<=t) two-tail	**0.010463114**	
t Critical two-tail	2.262157163	

Pada Tabel 7 dapat diketahui bahwa nilai rata-rata AUC dari model NB+GFS+BG lebih tinggi dibandingkan model NB sebesar 0.8093. Dalam uji beda statistik nilai alpha ditentukan sebesar 0.05, jika nilai p lebih kecil dibanding alpha ($p < 0.05$) maka H_0 ditolak dan H_1 diterima sehingga ada perbedaan yang signifikan antara model yang dibandingkan, akan tetapi jika nilai p lebih besar dibandingkan nilai alpha ($p > 0.05$) maka H_0 diterima dan H_1 ditolak sehingga tidak ada perbedaan yang signifikan antara model yang dibandingkan.

Pada Tabel 7 dapat dilihat bahwa nilai P(T<=t) adalah 0.01 dan ini menunjukkan bahwa nilai p lebih kecil dibandingkan nilai alpha (0.01 < 0.05) sehingga hipotesis H_0 ditolak dan H_1 diterima. Hipotesis H_1 diterima berarti ada perbedaan signifikan antara model NB dan model NB+GFS+BG sehingga model NB+GFS+BG membuat peningkatan ketika dibandingkan dengan model NB.

Selanjutnya dilakukan uji friedman untuk mengetahui model yang terbaik, pada Tabel 8 menunjukkan hasil p-value dari uji friedman. Nilai yang ditebalkan berarti nilai tersebut lebih kecil dari 0.05. Tabel 9 merupakan signifikansi dari model yang dibandingkan, model NB+GFS ada perbedaan yang signifikan dengan model NB dan model NB+GFS+BG ada perbedaan yang signifikan dengan model NB.

Tabel 8 P-Value Uji Friedman

	NB	NB+GFS	NB+GFS+BG
NB	1	**0.049**	**0.007**
NB+GFS	**0.049**	1	0.780
NB+GFS+BG	**0.007**	0.780	1

Tabel 9 Signifikan Model Uji Friedman

	NB	NB+GFS	NB+GFS+BG
NB	No	**Yes**	**Yes**
NB+GFS	**Yes**	No	No
NB+GFS+BG	**Yes**	No	No

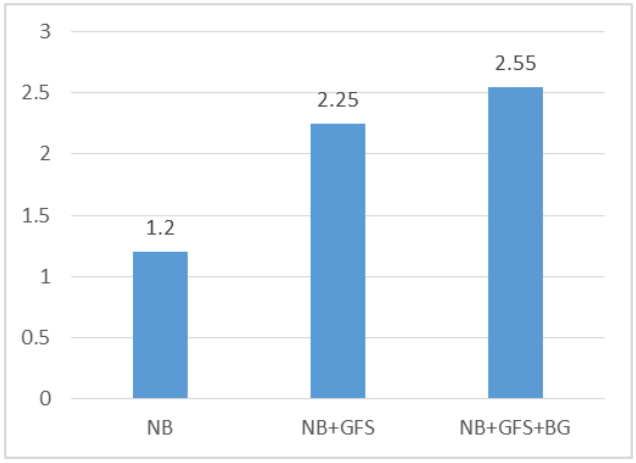
Tabel 10 Table of pairwise differences

	NB	NB+GFS	NB+GFS+BG
NB	0	**-1.050**	**-1.350**
NB+GFS	**1.050**	0	-0.300
NB+GFS+BG	**1.350**	0.300	0

Critical difference: 1,0481

Peringkat dari rata-rata diperoleh dari perbandingan model dan diketahui bahwa nilai peringkat rata-rata dibagi

dengan jumlah sampel yang digunakan. Perbedaan peringkat dari rata-rata masing-masing model dibandingkan dengan *Critical Difference* yang dapat dilihat pada Tabel 10. *Critical Difference* dapat dihitung menggunakan persamaan $CD = q_{a,\infty,L}\sqrt{\frac{L(L+1)}{12K}}$ dan dalam eksperimen ini dihasilkan nilai *Critical Difference* sebesar 1.0481. Jika nilai pada Tabel 10 lebih besar dibandingkan dengan nilai *Critical Difference* maka model tersebut mempunyai perbedaan yang signifikan dibandingkan model lainnya.



Gambar 5 Rata-Rata Peringkat Uji Friedman

Pada Gambar 5 merupakan rata-rata peringkat dari uji friedman, diketahui bahwa model NB+GFS+BG mempunyai rata-rata peringkat tertinggi, kemudian diikuti dengan model NB+GFS dan model NB mempunyai rata-rata peringkat terendah.

Selanjutnya untuk mengetahui model seleksi fitur yang terbaik dalam menangani fitur-fitur yang tidak relevan, dilakukan perbandingan model dengan seleksi fitur lain. Model yang dibandingkan yaitu: Naïve Bayes (NB), Naïve Bayes dan Bagging (NB+BG), Naïve Bayes dan Greedy Forward Selection (NB+GFS), Naïve Bayes dengan Greedy Forward Selection dan Bagging (NB+GFS+BG), Naïve Bayes dan Genetic Feature Selection (NB+GAFS), Naïve Bayes dan Forward Selection (NB+FS), Naïve Bayes dan Backward Elimination (NB+BE). Tabel 11 menunjukkan hasil AUC dari masing-masing model yang dibandingkan.

Tabel 11 Perbandingan AUC dengan Seleksi Fitur Lain

Model	CM1	JM1	KC1	KC3	MC2	PC1	PC2	PC3	PC4	PC5
NB	0.708	0.683	0.786	0.677	0.712	0.775	0.885	0.756	0.84	0.94
NB+BG	0.717	0.685	0.789	0.673	0.721	0.794	0.877	0.756	0.838	0.942
NB+GFS	0.769	0.653	0.788	**0.765**	0.775	0.81	0.894	0.784	0.86	0.941
NB+GFS+BG	**0.771**	**0.688**	0.788	0.763	**0.781**	**0.817**	**0.918**	0.778	0.866	0.923
NB+GAFS	0.757	0.635	0.79	0.719	0.758	0.79	0.75	0.792	0.857	**0.952**
NB+FS	0.601	0.612	**0.799**	0.749	0.707	0.742	0.824	0.583	0.812	0.886
NB+BE	0.717	0.659	0.768	0.734	0.275	0.799	0.908	**0.808**	**0.885**	0.938

Tabel 12 *Pairwise Differences*

	NB	NB+BG	NB+GFS	NB+GFS+BG	NB+GAFS	NB+FS	NB+BE
NB	0	-0.750	-2.200	-2.700	-1.250	0.750	-1.200
NB+BG	0.750	0	-1.450	-1.950	-0.500	1.500	-0.450
NB+GFS	2.200	1.450	0	-0.500	0.950	**2.950**	1.000
NB+GFS+BG	2.700	1.950	0.500	0	1.450	**3.450**	1.500
NB+GAFS	1.250	0.500	-0.950	-1.450	0	2.000	0.050
NB+FS	-0.750	-1.500	**-2.950**	**-3.450**	-2.000	0	-1.950
NB+BE	1.200	0.450	-1.000	-1.500	-0.050	1.950	0

Critical difference: 2,8483

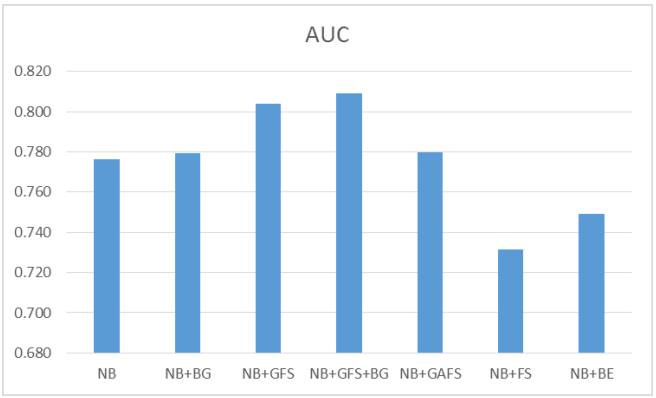
Pada Tabel 12 menunjukkan bahwa nilai NB+GFS dan NB+GFS+BG lebih besar dibandingkan nilai *Critical Difference* berarti model NB+GFS dan model NB+GFS+BG mempunyai perbedaan yang signifikan ketika dibandingkan dengan model NB+FS. Tabel 13 diketahui bahwa nilai p (*p-value*) lebih kecil dari nilai alpha sehingga ada perbedaan signifikan antara model. Pada Tabel 13 dapat dilihat bahwa model NB+GFS dan model NB+GFS+BG mempunyai perbedaan yang signifikan dengan model NB+FS sehingga hipotesa H_1 diterima dan kesimpulannya adalah model NB+GFS dan model NB+GFS+BG ada peningkatan ketika dibandingkan dengan model NB+FS. Model yang mempunyai perbedaan yang signifikan dan mempunyai peningkatan dibandingkan model lainnya ditandai dengan angka yang ditebalkan. Pada Tabel 14 merupakan hasil perbedaan signifikansi antara model yang dibandingkan, model yang berisi Yes merupakan model yang mempunyai perbedaan signifikan dengan model lainnya, sedangkan model yang berisi No merupakan model yang tidak mempunyai perbedaan signifikan terhadap model lainnya.

Tabel 13 Hasil P-Value Uji Friedman

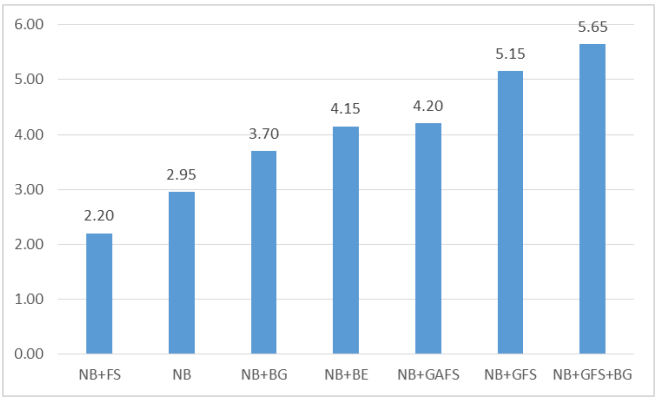
	NB	NB+BG	NB+GFS	NB+GFS+BG	NB+GAFS	NB+FS	NB+BE
NB	1	0.987	0.255	0.077	0.855	0.987	0.878
NB+BG	0.987	1	0.744	0.403	0.999	0.713	0.999
NB+GFS	0.255	0.744	1	0.999	0.958	**0.037**	0.946
NB+GFS+BG	0.077	0.403	0.999	1	0.744	**0.007**	0.713
NB+GAFS	0.855	0.999	0.958	0.744	1	0.371	1.000
NB+FS	0.987	0.713	**0.037**	**0.007**	0.371	1	0.403
NB+BE	0.878	0.999	0.946	0.713	1.000	0.403	1

Tabel 14 Perbedaan Signifikansi

	NB	NB+BG	NB+GFS	NB+GFS+BG	NB+GAFS	NB+FS	NB+BE
NB	No	No	No	No	No	No	No
NB+BG	No	No	No	No	No	No	No
NB+GFS	No	No	No	No	No	**Yes**	No
NB+GFS+BG	No	No	No	No	No	**Yes**	No
NB+GAFS	No	No	No	No	No	No	No
NB+FS	No	No	**Yes**	**Yes**	No	No	No
NB+BE	No	No	No	No	No	No	No



Gambar 6 Grafik AUC Mean Uji Friedman



Gambar 7 Grafik Rata-Rata Peringkat Uji Friedman

Pada Gambar 6 dapat dilihat bahwa model NB+GFS+BG merupakan model terbaik dalam penelitian ini dengan mempunyai rata rata AUC tertinggi dalam uji friedman

dan model NB+GFS merupakan model terbaik kedua yang diikuti oleh model NB+GAFS, NB+BG, NB, NB+BE dan NB+FS. Gambar 7 merupakan rata-rata peringkat dimana model NB+GFS+BG merupakan model terbaik yang menunjukkan bahwa seleksi fitur greedy forward selection dan bagging mampu meningkatkan kinerja model prediksi cacat *software* dengan memperbaiki model NB dan dapat menangani permasalahan ketidakseimbangan kelas (*imbalance class*) dan fitur-fitur yang tidak relevan (*irrelevant features*).

5. KESIMPULAN

Pengembangan *software* diperlukan agar menghasilkan *software* yang berkualitas. Kualitas *software* ditentukan dengan melakukan pemeriksaan dan pengujian, untuk menemukan cacat dalam *software* tersebut yang dapat menurunkan kualitas *software*. Dataset cacat *software* umumnya memiliki ketidaseimbangan kelas dan fitur-fitur yang tidak relevan yang dapat menyebabkan menurunnya kinerja dari algoritma pembelajaran. Salah satu algoritma pembelajaran yang digunakan dalam prediksi cacat *software* yaitu naïve bayes, yang merupakan algoritma terbaik untuk prediksi cacat atau tidaknya sebuah *software*. Pada penelitian ini mengusulkan model Naïve Bayes dan seleksi fitur Greedy Forward Selection (NB+GFS) untuk mengatasi permasalahan fitur-fitur yang tidak relevan, sedangkan untuk mengatasi permasalahan ketidakseimbangan kelas menggunakan model Naïve Bayes dengan Greedy Forward Selection dan Bagging (NB+GFS+BG). Hasil eksperimen dalam penelitian ini mendapatkan nilai AUC tertinggi pada model NB+GFS sebesar 0.941 dan untuk model NB+GFS+BG mendapatkan nilai AUC tertinggi sebesar 0.923 pada dataset PC5. Untuk menentukan model model terbaik dalam penelitian prediksi cacat *software*, selanjutnya dilakukan friedman test. Hasil friedman test diketahui bahwa nilai rata-rata peringkat dengan tertinggi adalah model NB+GFS+BG, sehingga model NB+GFS+BG merupakan model terbaik dalam penelitian prediksi cacat *software*.

REFERENSI

Arora, I., Tetarwal, V., & Saha, A. (2015). Open Issues in *Software* Defect Prediction. *Procedia Computer Science*, 906-912.

Chang, R., Mu, X., & Zhang, L. (2011). *Software* Defect Prediction Using Non-Negative Matrix Factorization. *Journal of Software*, 2114-2120.

Gao, K., Khoshgoftaar, T., & Wald , R. (2014). Combining Feature Selection and Ensemble Learning for *Software* Quality Estimation. *Twenty-Seventh International Florida Artificial Intelligence Research society Conference* (pp. 47-52). Association for the Advacement of Artificial Intelligence.

Gorunescu, F. (2011). *Data Mining Concepts, Models and Techniques.* Berlin: Springer.

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

Khoshgoftaar, T. M., Hulse, J. V., & Napolitano, A. (2011). Comparing Boosting and Bagging Techniques with Noisy and Imbalanced Data. *IEEE Transactions on Systems, Man and Cybernetics*, 552-568.

Laradji, I. H., Alshayeb, M., & Ghouti, L. (2015). *Software* Defect Prediction Using Ensemble Learning on Selected Features. *Information and Software Technology*, 388-402.

Lee, C.-H. (2015). A Gradient Approach for Value Weighted Classification Learning in *Naive Bayes*. *Knowledge-Based Systems*, 1-9.

Lessmann, S., Baesens, B., Mues, C., & Pietsch, S. (2008). Benchmarking Classification Models for *Software* Defect Prediction: A Proposed Framework and Novel Findings. *IEEE Transactions on Software Engineering*, 485-496.

Liu, X.-Y., & Zhou, Z.-H. (2013). Ensemble Methods for Class Imbalance Learning. *Imbalanced Learning: Foundations, Algorithms, and Applications, First Edition*, 61-82.

Ma, Y., Luo, G., Zeng, X., & Chen, A. (2012). Transfer Learning for Cross-Company *Software* Defect Prediction. *Information and Software Technology*, 248-256.

Song, Q., Jia, Z., Shepperd, M., Ying, S., & Liu, J. (2010). A General *Software* Defect-Proneness Prediction Framework. *IEEE Transaction on Software Engineering*, 1-16.

Strate, J. D., & Laplante, P. A. (2013). A Literature Review of Research in *Software* Defect Reporting. *IEEE Transactions on Reliability*, 444-454.

Wahono, R. S. (2015). A Systematic Literature Review of *Software* Defect Prediction: Research Trends, Datasets, Methods and Frameworks. *Journal of Software Engineering*, 1-16.

Wahono, R. S., & Suryana, N. (2013). Combining Particle Swarm Optimization based Feature Selection and Bagging Technique for *Software* Defect. *IJSEIA*, 153-166.

Wahono, R. S., Suryana, N., & Ahmad, S. (2014). Metaheuristic Optimization based Feature Selection for *Software* Defect Prediction. *Journal of Software*, 1324-1333.

Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining Practical Machine Learning Tools and techniques.* Burlington: Elsevier.

BIOGRAFI PENULIS

Fitriyani. Memperoleh gelar S.T pada bidang Sistem Informasi dari Universitas BSI Bandung dan gelar M.Kom pada bidang *software engineering* dari STMIK Nusa Mandiri. Staf pengajar di Universitas BSI Bandung. Minat penelitian saat ini meliputi *software engineering* dan *machine learning*.

Romi Satria Wahono. Memperoleh gelar B.Eng. dan M.Eng. di bidang Ilmu Komputer dari Saitama University, Japan, dan gelar Ph.D. di bidang Software Engineering dari Universiti Teknikal Malaysia Melaka. Dia saat ini sebagai dosen program Pascasarjana Ilmu Komputer di Universitas Dian Nuswantoro, Indonesia. Dia juga pendiri dan CEO PT Brainmatics Cipta Informatika, sebuah perusahaan pengembangan perangkat lunak di Indonesia. Minat penelitiannya saat ini meliputi rekayasa perangkat lunak dan machine learning. Anggota Profesional ACM dan IEEE Computer Society.