# Komparasi Algoritma Logistic Regression dan Random Forest pada Prediksi Cacat Software

**Rizal Prasetyo[1], Imam Nawawi[2], Ahmad Fauzi[3], Ginabila[4]**
[1]Universitas Nusa Mandiri; Jalan Jatiwaringin - Jakarta Timur
[2,3,4]Universitas Bina Sarana Informatika; Jalan Kramat Raya - Jakarta Selatan
Email: [1]14002272@nusamandiri.ac.id, [2]imam.imw@bsi.ac.id , [3]ahmad.aau@bsi.ac.id,
[4]gina.gnb@bsi.ac.id

***Abstrak***

*Pengujian menjadi strandar dalam menghasilkan perangkat lunak yang berkulaitas, pengujian dapat dinilai melalui ukuran-ukuran dan metode-metode tertentu, salah satu tolak ukur kualitas perangkat lunak adalah ISO, yang dibuat oleh International Organization for Standardization (ISO) dan International Electrotechnical Commission (IEC), Dalam prediksi cacat perangkat lunak kesalahan prediksi cacat perangkat lunak merupakan hal yang sangat buruk, hasil prediksi dapat menimbulkan pengaruh terhadap perangkat lunak itu sendiri. Penelitian ini membandingkan hasil kinerja Algoritma Logistic Regression dan Random Forest sebelum dan setelah diterapkan metode resampling, hasil pengujian menujukan bahwa Random Forest dengan resampling menghasilkan tingkat akurasi yang lebih tinggi. Dari hasil pengujian di atas dapat disimpulkan bahwa Random Forest dengan metode resampling lebih efektif pada prediksi cacat software.*

***Kata Kunci**: Logistic Regression, Random Forest, Resampling, rediksi cacat software*

***Abstract***

*Testing becomes the standard in producing quality software, testing can be assessed through certain measures and methods, one of the benchmarks for software quality is ISO, which was made by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) , In the prediction of software defects software defect prediction error is a very bad thing, the prediction results can have an effect on the software itself. This study compares the results of the Logistic Regression Algorithm and Random Forest before and after the resampling method is applied, the test results show that Random Forest with resampling produces a higher level of accuracy. From the test results above, it can be concluded that the Random Forest with resampling method is more effective in predicting software defects.*

***Keywords**: Logistic Regression, Random Forest, Resampling, software defect prediction*

## 1. PENDAHULUAN

Salah satu ciri *software* berkualitas adalah yang paling minimal memiliki kecenderungan cacat (Akbar and Rochimah 2017). Prediksi cacat *software* merupakan salah satu cara yang efektif untuk meminimalkan biaya dan usaha dalam proses penggunaan dan perbaikan dalam membangun atau mengembangkan *software* (Fauzi and Ginabila 2019).

Atribut kode statis atau biasa disebut *software metric* diekstraksi dari *software* untuk memprediksi modul yang rusak. Prediktor atau modul yang terdeteksi memiliki kecenderungan cacat diharapkan dapat membantu meningkatkan kualitas *software* untuk dirilis pada periode berikutnya (Wang and Yao 2013). Berbagai *software metrics* yang berbeda ditemukan setelah proses ekstraksi *software* dan digunakan untuk memprediksi kecenderungan cacat. Untuk mempermudah dan membuat proses prediksi kecacatan *software* lebih praktis, dipilih set *metrics* yang paling penting dari semua *software metrics* yang ada (Okutan and Yildiz 2014).

Keseimbangan kelas atau *class imbalanced* merupakan hal yang sangat penting dalam rangka menghasilkan data training atau *training set* yang baik. Hampir semua algoritma klasifikasi menunjukkan performa yang sangat buruk ketika bekerja pada data dengan kelas yang sangat tidak seimbang atau imbalance. Pada hakekatnya data real adalah tidak seimbang (Siringoringo 2017).

Pada penelitian sebelumnya dilakukan oleh (Aries and Wahono 2015). melakukan melakukan pengujian terhadap ketidakseimbanagan kelas diatasi dengan berbagai pendekatan level data yaitu *Random Undersampling*, *Random Oversampling* dan FSMOTE dengan algoritma Naive Bayes. Hasil dari penelitian tersebut pendekatan level data FSMOTE dengan algoritma Naive Bayes menghasilkan tingkat akurasi yang paling tinggi untuk mengatasi ketidakseimbangan kelas data. Penelitian lain yang pernah di lakukan oleh (Fauzi and Ginabila 2019) melakukan komparasi dengan pendekatan level data *Random Undersampling* (RUS) dengan mengambil beberapa algoritma Naive Bayes (NB), J48 dan Random Forest (RF) untuk membandingkan mana tingkat akurasi yang lebih tinggi, dan didapatkan presentase 71,932%. dari algoritma Random Forest setelah dilakukan pendekatan *Random Undersampling*, memiliki hasil tertinggi di antara Algoritma yang lainnya

Dari hasil penelitian sebelumnya yang telah dilakukan maka peneliti melakukan penelitian lanjutan dengan meggunakan metode *resampling* dan komparasi algoritma Random Forest (RF) dan Logistic Regression (LR) dengan data yang berbeda dari dua penelitian yang telah dilakukan. Oleh tujuannya yaitu mengetahui nilai akurasi mana yang paling tinggi dan dapat memberikan hasil yang lebih maksimal.

## 2. BAHAN DAN METODE

Logistic Regression merupakan klasifikasi linier yang menangani masalah klasifikasi multi kelas dan telah terbukti menghasilkan klasifikasi yang *powerful*, (Sperandei 2014) Ketidakseimbangan kelas menjadi masalah besar yang dialami oleh algoritma Logistic Regression (class imbalance) pada dataset berdimensi tinggi (Field 2012).

Random forest merupakan pengembangan dari Algoritma C4.5 (*decision tree*) dengan menggunakan beberapa *decision tree*, dimana setiap decision tree telah dilakukan training data menggunakan sampel individu dan setiap atribut dipecah pada *tree* yang dipilih antara atribut subset yang bersifat acak. Dan dalam perkembangannya, sejalan dengan bertambahnya dataset, maka *tree* pun ikut berkembang (Frastian, Hendrian, and Valentino 2018).

*Random undersampling* yaitu menghasilkan subsampel acak dari instance kelas mayoritas *Undersumpling*: metode *resampling* ini secara acak memilih sampel di kelas mayoritas dan menambahkannya ke kelas minoritas, membentuk sebuah dataset pelatihan baru (Rajesh and Dhuli 2018).

Dalam penelitian ini dilakukan komparasi dua algoritma pengklasifikasi yaitu Random forest (RF) dan Logistic Regression (LR). Komparasi dua algoritma ini menggunakan metode *resampling* untuk mengatasi ketidak seimbangan kelas dari data yang diambil dari suatu repositori.

### 2.1. Pengumpulan Data

Penulis menggunakan data *software defect* yang telah tersedia dari PROMISE repositori, diambil dari situs http://promise.site.uottawa.ca. Data tersebut berisi 22 atribut *software metrics* dan 498 baris data. Atribut-Atribut yang terdapat dalam data tersebut adalah:

**Tabel 1**. Atribut Dalam Data

| No | Atribut |
|---|---|
| 1 | loc |
| 2 | v(g) |
| 3 | ev(g) |
| 4 | iv(g) |
| 5 | n |
| 6 | v |
| 7 | l |
| 8 | d |
| 9 | i |

| No | Atribut |
|---|---|
| 10 | e |
| 11 | b |
| 12 | t |
| 13 | lOCode |
| 14 | lOComment |
| 15 | lOBlank |
| 16 | locCodeAndComment |
| 17 | uniq_Op |
| 18 | uniq_Opnd |
| 19 | total_Op |
| 20 | total_Opnd |
| 21 | branchCount |
| 22 | defects |

## 2.2. Pengolahan Data

Dalam proses pengolahan data, terdapat 2 tahap yang dilakukan oleh peneliti yaitu:

a. Proses Pengolahan Data Asli
   Dalam tahap ini peneliti mengkomparasikan 3 algoritma pengklasifikasi yang diambil dalam penelitian dengan data awal yang didapatkan dari PROMISE repositori tanpa digunakan pendekatan level data *Random Undersampling*,

b. Proses Pengolahan Ketidakseimbangan Kelas
   Pada tahap ini 3 algoritma pengklasifikasi digunakan dengan pendekatan level data *Random Undersampling* untuk mengetahui tingkat akurasi masing-masing algoritma setelah masalah ketidakseimbangan kelasnya diatasi.

Nilai pada masing-masing kelas dan rata-rata dari kedua kelas yang dibutuhkan untuk mengetahui keakuratan prediksi cacat *software* akan dihitung dan didapatkan tingkat akurasinya:

- TP Rate (True Positive Rate)

  $$TPR = \frac{TP}{TP+FN} \quad \text{..... (1)}$$

  Keterangan:
  TPR = *True Positive Rate*
  TP = *True Positive*
  FN = *False Negative*

- FP Rate (False Positive Rate)

  $$FPR = \frac{FP}{FP+TN} \quad \text{..... (2)}$$

  Keterangan:
  FPR = *False Positive Rate*
  FP = *False Positive*
  TN = *True Negative*

- Precision

  $$Precision = \frac{TP}{TP+FP} \quad \text{..... (3)}$$

  Keterangan:
  TP = *True Positive*
  FP = *False Positive*

- Recall

  $$Recall = \frac{TP}{TP+FN} \quad \text{..... (4)}$$

Keterangan:
TP = *True Positive*
FN = *False Negative*

- F-Measure

  Recall = $2 . \frac{Precision . Recall}{Precision+Recall}$ ..... (5)

- MCC

  MCC = $\frac{TP*TN-FP*FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$ ..... (6)

  Keterangan:
  TP = *True Positive*
  TN = *True Negative*
  FP = *False Positive*
  FN = *False Negative*

- ROC Area

  Kurva ROC adalah grafik dua dimensi dimana nilai TPR (True Positive Rate) mewakili sumbu Y dan nilai FPR (False Positive Rate) mewakili sumbu X (Tharwat 2018). Semakin dekat garis dengan angka 1 maka akan semakin baik. Pada bagian hasil dan pembahasan juga akan ditampilkan grafik ROC area dari masing masing kelas.

- PRC Area

  *Precision Recal Curve* biasanya digunakan jika ditemui kasus dimana data memiliki kelas yang tidak seimbang. Sesuai dengan namanya Kurva ini dibuat berdasarkan nilai yang telah didapatkan pada perhitungan dengan *confusion matrix*, yaitu antara *Precision* dan *Recall*.

- Accuracy

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

  Keterangan:
  TP = *True Positive*
  TN = *True Negative*
  FP = *False Positive*
  FN = *False Negative*

## 3. HASIL DAN PEMBAHASAN

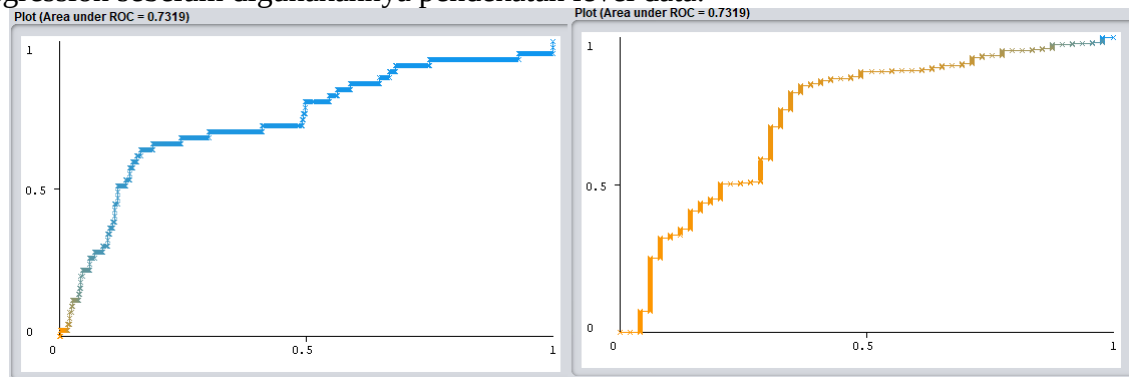
### 3.1. Hasil Perhitungan Menggunakan Algoritma Logistic Regression dan Random Forest Sebelum Menggunakan *resampling*

Dari 498 data *software defect* yang ada, dengan menggunakan algoritma pengklasifikasi Logistic Regression dan Random Forest didapatkan hasil perhitungan sebagai berikut:

**Tabel 2**. Hasil Perhitungan Logistic Regression

| | Class False | Class True | Rate |
|---|---|---|---|
| TP Rate | 0.967 | 0.122 | 0.884 |
| FP Rate | 0.878 | 0.033 | 0.794 |
| Precision | 0.910 | 0.286 | 0.848 |
| Recall | 0.967 | 0.122 | 0.884 |
| F-Measure | 0.937 | 0.171 | 0.862 |
| MCC | 0.132 | 0.132 | 0.132 |
| ROC Area | 0.732 | 0.732 | 0.732 |
| PRC Area | 0.942 | 0.245 | 0.873 |

Berikut adalah grafik ROC dari masing masing kelas dengan algoritma Logistic Regression sebelum digunakannya pendekatan level data:



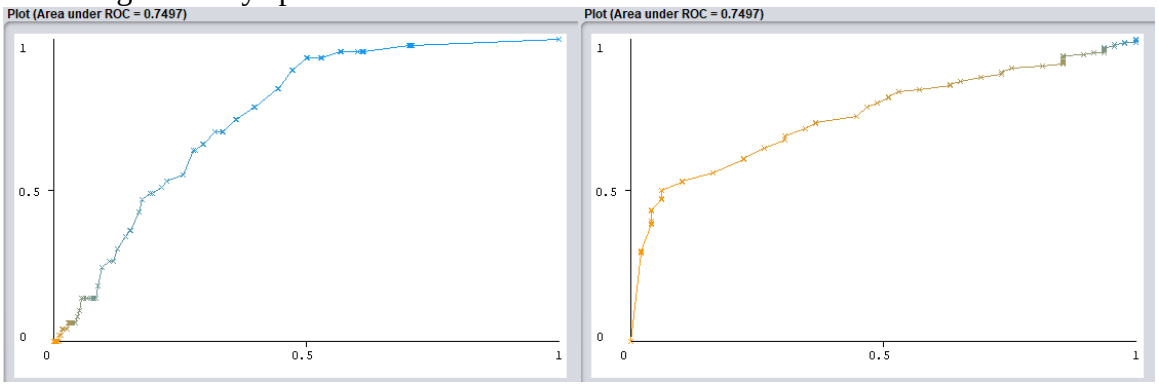

Gambar 1. ROC Class False & Class True Algoritma Logistic Regression

Dari Gambar 1 diatas dapat dilihat bahwa *Area Under ROC* dari *class true* dengan menggunakan algoritma Logistic Regression menunjukkan hasil 0,73. Sedangkan *Area Under ROC* dari *class false* juga menunjukkan hasil 0,73. Hasil tersebut menghasilkan kategori yang baik dikarenakan angka yang dihasilkan dari *Area Under ROC* mendekati angka 1.

**Tabel 3.** Hasil Perhitungan RandomForest

| | Class False | Class True | Rate |
|---|---|---|---|
| **TP Rate** | 0.980 | 0.041 | 0.888 |
| **FP Rate** | 0.959 | 0.020 | 0.867 |
| **Precision** | 0.903 | 0.182 | 0.832 |
| **Recall** | 0.980 | 0.041 | 0.888 |
| **F-Measure** | 0.940 | 0.067 | 0.854 |
| **MCC** | 0.042 | 0.042 | 0.042 |
| **ROC Area** | 0.750 | 0.750 | 0.750 |
| **PRC Area** | 0.964 | 0.191 | 0.888 |

Berikut adalah grafik ROC dari masing masing kelas dengan algoritma RandomForest sebelum digunakannya pendekatan level data:




**Gambar 2.** ROC Class False & Class True Algoritma RandomForest

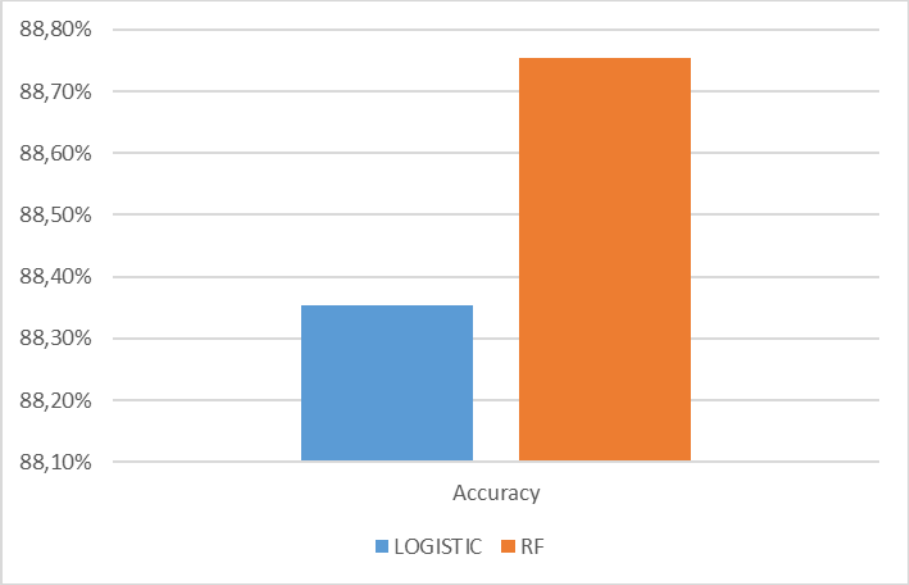
Dari Gambar 2 diatas dapat dilihat bahwa *Area Under ROC* dari *class true* dengan menggunakan algoritma Random Forest menunjukkan hasil 0,743. Sedangkan *Area Under ROC* dari *class false* juga menunjukkan hasil 0,74. Hasil tersebut juga menghasilkan kategori yang baik dikarenakan angka yang dihasilkan dari *Area Under ROC* mendekati angka 1.

**Tabel 4.** Hasil Perhitungan Akurasi LR dan RF

| | LR | RF |
|---|---|---|

| | | |
|---|---|---|
| Correctly Classified Instances | 88.353% | 88.755% |
| Incorrectly Classified Instances | 11.646% | 11.245% |

Berikut adalah grafik perbandingan tingkat akurasi sebelum digunakan pendekatan level data Random Undersampling dari masing-masing algoritma:




**Gambar 3.** Perbandingan Tingkat Akurasi dari Kedua Algoritma

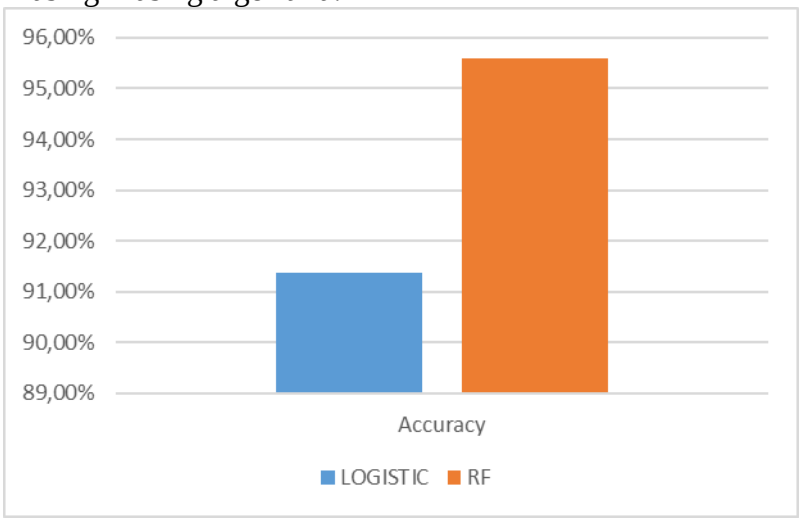
Grafik diatas menampilkan hasil akurasi yang didapatkan dari penggunaan algoritma Random Forest lebih tinggi dari Algoritma Logistic Regression sebelum kelas nya diseimbangkan menggunakan Random Undersampling. Algoritma Logistic Regression menghasilkan akurasi 88,35%, sedangkan Algoritma Random Forest menghasilkan 88,75%.

Setelah melalui perhitungan yang dilakukan, dapat diketahui nilai akurasi dari masing masing algoritma setelah diterapkan metode resampling adalah sebagai berikut:

**Tabel 5.** Hasil Perhitungan Akurasi LR dan RF

| | LR | RF |
|---|---|---|
| Correctly Classified Instances | 91.365% | 95.582% |
| Incorrectly Classified Instances | 8.634% | 4.417% |

Berikut adalah grafik perbandingan tingkat akurasi setelah digunakan diterapkan metode resampling dari masing-masing algoritma:



**Gambar 4.** Perbandingan Tingkat Akurasi dari Kedua Algoritma

Grafik diatas menampilkan hasil akurasi yang didapatkan dari penggunaan Algoritma Logistic Regression dan Random Forest setelah kelas data diseimbangkan menggunakan Random Undersampling. Setelah data diseimbangkan hasil akurasi yang didapatkan kedua algoritma meningkat pesat. Hasil akurasi yang didapatkan Algoritma Random Forest tetap menghasilkan akurasi yang lebih tinggi. Algoritma Logistic Regression menunjukkan akurasi 91,4%, sedangkan Algoritma Random Forest menghasilkan 95,6%.

## 4. KESIMPULAN

Dari hasil komparasi yang dilakukan dengan Algoritma Logistic Regression dan Random Forest sebelum dan sesudah menggunakan metode *resampling*, kedua Algoritma meiliki peningkatan akurasi setelah menggunakan *resampling*, dan tingkat akurasi yang paling tinggi adalah dengan menggunakan algoritma Random Forest dan *resampling* dengan presentase 95.582%.

## DAFTAR PUSTAKA

[1]. Akbar, Muhammad Sonhaji, and Siti Rochimah. 2017. "Prediksi Cacat Perangkat Lunak Dengan Optimasi Naive Bayes Menggunakan Gain Ratio." *Jurnal Sistem dan Informatika*: 147–55. http://repository.its.ac.id/2527/.

[2]. Aries, Saifudin, and Romi Satria Wahono. 2015. "Pendekatan Level Data Untuk Menangani Ketidakseimbangan Kelas Pada Prediksi Cacat Software." *Journal of Software Engineering* 1(2): 76–85.

[3]. Fauzi, Ahmad, and Ginabila. 2019. "Komparasi Algoritma Dengan Pendekatan Random." 15(1): 27–34.

[4]. Field, Andy. 2012. "Logistic Regression Logistic Regression Logistic Regression." *Discovering Statistics Using SPSS*: 731–35.

[5]. Frastian, Nahot, Senna Hendrian, and V.H. Valentino. 2018. "Komparasi Algoritma Klasifikasi Menentukan Kelulusan Mata Kuliah Pada Universitas." *Faktor Exacta* 11(1): 66.

[6]. Okutan, Ahmet, and Olcay Taner Yildiz. 2014. "Software Defect Prediction Using Bayesian Networks." *Empirical Software Engineering* 19(1): 154–81.

[7]. Rajesh, Kandala N.V.P.S., and Ravindra Dhuli. 2018. "Classification of Imbalanced ECG Beats Using Re-Sampling Techniques and AdaBoost Ensemble Classifier." *Biomedical Signal Processing and Control* 41(March): 242–54. http://dx.doi.org/10.1016/j.bspc.2017.12.004.

[8]. Siringoringo, Rimbun. 2017. "Integrasi Metode Resampling Dan K-Nearest Naighbor Pada Prediksi Cacat Software Aplikasi Android." *ISD* 2 No.1(1): 47–58.

[9]. Sperandei, Sandro. 2014. "Understanding Logistic Regression Analysis." *Biochemia Medica* 24(1): 12–18.

[10]. Tharwat, Alaa. 2018. "Classification Assessment Methods." *Applied Computing and Informatics*.

[11]. Wang, Shuo, and Xin Yao. 2013. "Using Class Imbalance Learning for Software Defect Prediction." *IEEE Transactions on Reliability* 62(2): 434–43.