HASIL_60150823

by Pgsd 60150823

Submission date: 13-Feb-2023 03:14PM (UTC+0700)
Submission ID: 2012975739
File name: PGSD-60150823-JURNAL_4_-_Mukti_Sintawati.doc (227.5K)
Word count: 3013
Character count: 18281

JUSTINDO
Jurnal Sistem dan Teknologi Informasi
http://jurnal.unmuhjember.ac.id/index.php/JUSTINDO
E-ISSN: 25541-5735, P-ISSN: 2502-5724
Volume 7, No. 1, Februari 2022

Klasifikasi Penyakit Diabetes Melitus Menggunakan *Adaboost Classifier*

Ginanjar Abdurrahman

Universitas Muhammadiyah Jember
Email: abdurrahmanginanjar@unmuhjember.ac.id

(Naskah masuk: 22 April 2021, diterima untuk diterbitkan: 5 Februari 2022, Terbit: 28 Februari 2022)

ABSTRAK

Diabetes Melitus (DM) merupakan penyakit dengan gejala kadar gula darah sewaktu lebih dari 200 mg/dL, dan kadar gula darah puasa lebih dari 126 mg/dL. Klasifikasi merupakan algoritma untuk pencarian pola dengan membangun model klasifikasi berdasarkan variabel kelas prediktor dan variabel kelas target. *Adaboost (Adaptive Boosting)* merupakan salah satu algoritma klasifikasi *(classifier)* yang dapat membangun *strong classifier* dengan menggabungkan beberapa *weak classifier.* Algoritma ini juga dapat menyesuaikan diri dengan data dan metode *classifier* yang lain. Kelebihan lain dari algoritma ini adalah dapat memperkecil tingkat *error* dari *weak classifier* sehingga dapat menaikkan tingkat akurasi dari algoritma pembelajaran yang ada (*boosting).* Pada penelitian ini akan dilakukan klasifikasi penyakit DM menggunakan algoritma *Adaboost Classifier* untuk menentukan apakah seseorang menderita diabetes atau tidak. Dataset diperoleh dari *UCI Machine Learning*, dengan 8 variabel kelas prediktor, 1 variabel kelas target, serta 768 *record*. Hasil klasifikasi *Adaboost Clasifier* pada dataset setelah *imputing mean* diperoleh nilai akurasi sebesar 80.09 %, sedangkan untuk dataset setelah *imputing median* diperoleh nilai akurasi sebesar 76.19 %, untuk dataset setelah *imputing* modus, diperoleh hasil yang sama dengan *default dataset* yang belum dilakukan *imputing*, akibatnya *Adaboost classifier* tidak bisa berjalan karena *Adaboost* sangat sensitif terhadap *missing values*. Nilai *missing values* untuk beberapa fitur paling sering muncul, dikenali sebagai modus oleh *python* sehingga nilai *missing values* digantikan dengan *NaN*.

Kata kunci: prediktor, *classifier*, *adaboost classifier*, *imputing*, *missing values*

ABSTRACT

Diabetes mellitus (DM) is a disease with symptoms when blood sugar levels are more than 200 mg/dL, and fasting blood sugar levels are more than 126 mg/dL. Classification is an algorithm for pattern search by building a classification model based on predictor class variables and target class variables. Adaboost (Adaptive boosting) is a classification algorithm (classifier) that can build a strong classifier by combining several weak classifier. This algorithm can also adapt to other data and classifier methods. Another advantage of this algorithm is that it can reduce the error rate of the weak classifier so that it can increase the level of accuracy of the existing learning agorithm (boosting). In this study, DM will be classified using the Adaboost Classifier algorithm to determine whether a person has diabetes or not. The dataset was obtained from UCI Machine Learning, with 8 predictor class variables, 1 target class variable, and 768 records. The results of the Adaboost Classifier classification on the dataset after imputing the mean obtained an accuracy value of 80.09%, while for the dataset after the median imputing an accuracy value of 76.19% was obtained, for the dataset after the

imputing mode, the same results were obtained as the default dataset that had not been imputed, as a result Adaboost classifier cannot run because Adaboost is very sensitive to missing values. Missing values for some features occur most frequently, recognized as mode by python so missing values are replaced with NaN.

Keywords: predictor, *classifier*, *adaboost classifier*, *imputing*, *missing values*

1. PENDAHULUAN

Diabetes melitus merupakan penyakit dengan gejala kadar gula darah sewaktu lebih dari 200 mg/dL, dan kadar gula darah puasa lebih dari 126 mg/dL (Misnadiarly(Hestiana, 2017)). International Diabetes Federation (IDF) menjadikan Diabetes sebagai penyakit paling mamatikan urutan ke-tujuh di dunia dengan prevalensi 1.9%. Pada tahun 2013, penderita diabetes dunia mencapai 382 juta jiwa, dengan 95% diantaranya adalah DM tipe 2.

Machine learning merupakan suatu teknik untuk meniru cara mesin dalam "belajar" dari data *(learn from data)* (Lukman & Marwana, 2014). Python merupakan Bahasa pemrograman untuk data *analyst*, data *scientist*, juga data *engineer* dalam implementasi machine learning (Purwadhika, 2019).

Klasifikasi merupakan algoritma untuk pencarian pola dengan membangun model klasifikasi berdasarkan variabel kelas prediktor dan variabel kelas target (Bimo et al., 2020).

Adaboost (Adaptive Boosting) merupakan salah satu algoritma klasifikasi yang ditemukan Yoav Freund dan Robert Schapire. Algoritma ini membangun *strong classifier* dengan menggabungkan beberapa *weak classifier.* Algoritma ini juga dapat menyesuaikan diri dengan data dan algoritma *classifier* lainnya, sehingga disebut *adaptive*. Selain itu, algoritma ini juga dapat memperkecil *error* dari *weak classifier* sehingga dapat menaikkan akurasi dari setiap algoritma pembelajaran, sehingga algoritma ini bernama *boosting* (Imaduddin & Tawakal, 2015).

Pada penelitian ini akan mengklasifikasikan penyakit DM dengan algoritma *Adaboost Classifier* untuk mengklasifikasikan seseorang sebagai penderita diabetes atau tidak. Dipilihnya algoritma *Adaboost* dalam penelitian ini dikarenakan kelebihan *AdaBoost* dalam membangun *strong classifier* dengan menggabungkan *weak classifier*, Algoritma ini dapat beradaptasi dengan data dan metode *classifier* lainnya. Selain itu, algoritma ini juga dapat memperkecil tingkat *error* dari *weak classifier* sehingga dapat menaikkan akurasi dari algoritma pembelajaran yang ada. Dataset dalam penelitian adalah dataset diabetes dari repositori UCI. Dataset ini terdiri dari 8 variabel kelas prediktor, 1 variabel kelas target, serta 768 *record*.

2. PENELITIAN TERKAIT

Penelitian yang dilakukan (Imaduddin & Tawakal, 2015) untuk mendeteksi dan mengklasifikasikan daun secara otomatis menggunakan algoritma *Adaboost* dan *SVM*. Pada penelitian ini dilakukan deteksi daun otomatis sekaligus mengenali objek daun. Metode *AdaBoost* digunakan untuk deteksi otomatis letak dan jenis daun. Performa deteksi *AdaBoost* dari penelitian ini memiliki akurasi 84,23 %, sedangkan klasifikasi dengan *SVM* memiliki akurasi 71 %.

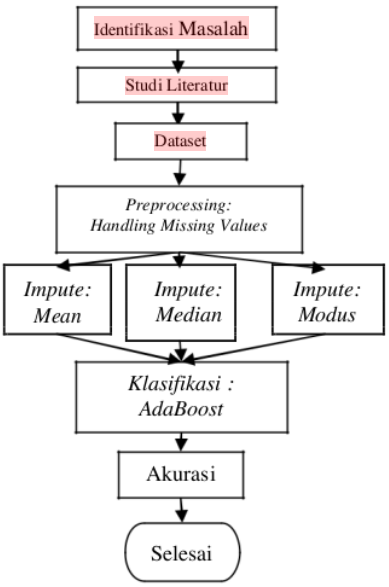
p-ISSN : 2502-5724; e-ISSN : 2541-5735

Penelitian selanjutnya (Oktanisa & Supianto, 2018) bertujuan membandingkan kinerja 9 algoritma klasifikasi (*SVM, Adaboost, Naïve Bayes, Constant, KNN, Tree, Random Forest, SGD, dan CN2 Rule*) terhadap tanggapan nasabah. *Preprocessing* yang dilakukan adalah menghapus *missing value* dan ekstraksi fitur dari dataset. Pada tahap evaluasi dilakukan Teknik 10 *fold cross validation*. Setelah diuji, diperoleh hasil klasifikasi terbaik adalah model Tree dengan CA 0.97, *precission* 0.95, dan *recall* 0.98.

Selain sebagai classifier, *Adaboost* juga dapat digunakan untuk meningkatkan kinerja algoritma *classifier*. Seperti penelitian (Rohman et al., 2017) dengan tujuan memprediksi penyakit jantung dengan algoritma *C4.5*. Performa algoritma *C4.5* ditingkatkan dengan Algoritma AdaBoost yang diimplementasikan pada data penderita penyakit jantung. Berdasarkan confusion matrix dan kurva *ROC*, metode *C45* berbasis Adaboost diperoleh akurasi 86.6%, nilai *AUC* yang diperoleh 0.96 dan setelah dipotimasi dengan algoritma *Adaboost* nilai akurasinya menjadi 92,24% nilai AUC 0.982.

Berdasarkan beberapa penelitian sebelumnya, pada umumnya *Adaboost* digunakan untuk meningkatkan performa algoritma klasifikasi karena merupakan algoritma *boosting*. Akan tetapi, dalam penelitian ini algoritma *Adaboost* akan digunakan sebagai *classiifier (*algoritma klasifikasi). *Adaboost* dapat digunakan sebagai *classifier*, karena struktur dasar algoritma ini adalah pohon keputusan (Kurniawati, 2021). Performa algoritma klasifikasi *adaboost,* yakni akurasi algoritma akan ditinjau berdasarkan algoritma *imputing missing value* yang diterapkan, dalam hal ini *imputing* dengan menggunakan nilai mean, median, dan modus.

3. METODE PENELITIAN



3.1 Identifikasi Masalah

Diabetes melitus (DM) merupakan penyakit yang memiliki dampak terhadap kualitas hidup dan perekonomian. Prevalensi yang tinggi dari penyakit ini menjadi penyebab kematian urutan ketujuh di dunia. Penyakit DM perlu dikenali sejak awal, sehingga dapat ditangani sesegera mungkin. Untuk itulah, perlu adanya algoritma klasifikasi untuk mengenali penyakit DM secara akurat. Algoritma *Adaboost (Adaptive Boosting)* merupakan salah satu alternatif algoritma klasifikasi yang ditawarkan dalam penelitian ini. Algoritma ini dipilih karena dapat membangun *strong classifier* dengan menggabungkan beberapa *weak classifier*. Selain itu, algoritma ini bisa beradaptasi *(adaptive)* dengan data dan algoritma *classifier* lainnya. Algoritma ini juga dapat memperkecil tingkat *error* dari *weak classifier* sehingga dapat meningkatkan akurasi *(boosting)* dari setiap algoritma pembelajaran yang digunakan (Imaduddin & Tawakal, 2015).

3.2 Studi Literatur

Studi literatur merupakan langkah untuk mempelajari referensi berupa jurnal penelitian, buku-buku referensi relevan dengan penelitian.

3.3 Dataset

Dataset pada penelitian ini diperoleh dari *UCI Machine Learning,* yakni pima Indian diabetes. Data yang digunakan adalah data pasien diabetes yang berasal dari *National Institute of Diabetes and Kidney Diseases*. Semua pasien adalah wanita setidaknya berusia 21 tahun dari keturunan India Pima. Dataset ini terdiri dari 8 variabel input yakni: Kehamilan, Glukosa, Tekanan Darah, Ketebalan Kulit, Insulin, BMI, fungsi silsilah diabetes, dan umur. Selain itu, juga ada 1 variabel target, yang terdiri dari dua kelas keputusan: Penderita (1), Bukan Penderita (0). Dataset ini terdiri dari 768 *record,*

3.4 *Preprocessing*

Preprocessing yang dilakukan hanya penanganan *missing values* saja. Hal ini dikarenakan, gangguan data yang ada hanyalah *missing values*. Dari dataset yang digunakan, untuk fitur kelas keputusan sudah dalam bentuk *boolean* nol dan satu, bukan dalam karakter *string* berupa kata, sehingga tidak perlu dilakukan *binarization*. Dari dataset yang digunakan, terdapat *missing values* dari bebarapa fitur, yakni pada fitur: *Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, dan BMI.*

3.5 Klasifikasi *Adaboost*

Menurut Yoav & Robert (Mulyati et al., 2017) *Adaboost (Adaptive Boosting)* merupakan algoritma *machine learning* yang dirumuskan oleh Yoav Freund dan Robert Schapire. Persamaan *AdaBoost* dirumuskan sebagai:

Pada persamaan (1), merupakan based classifier, merupakan tingkat pembelajaran (*learning rate*), merupakan *strong classifier*

Menurut Zhou & Yu (Mulyati et al., 2017) Langkah-langkah Algoritma *Adaboost* adalah sebagai berikut:

Masukan:

Algoritma *Weak Learner*

L;

integer T yang menspesifikasi banyaknya iterasi

Proses

Inisialisasi bobot distribusi

untuk semua i = 1, 2, … , m

for t = 1, …, T:

Melatih weak learner dari D dengan distribusi

Menghitung kesalahan dari

If

Penetapan bobot dari

Memperbarui distribusi, dengan adalah faktor normalisasi untuk

menggenerate menjadi distribusi

end

Keluaran:

Strong classifier:

p-ISSN : 2502-5724; e-ISSN : 2541-5735

3.6 Akurasi

Menurut Olson & Young (Defiyanti, 2015) *confusion matrix* merupakan metode yang menggunakan table matriks yang dapat digunakan apabila kelas keputusan pada suatu dataset hanya terdiri dari dua kelas, yakni satu kelas dianggap positif dan kelas yang lain dianggap sebagai negative. Tabel *confusion matriks* dapat dilihat pada Tabel 2.

Tabel 1 *Confusion Matrix*

		True Class	
		Positive	*Negative*
Predicted Class	*Positive*	*TP*	*FP*
	Negative	*FN*	*TN*

Gorunescu (Bisri, 2015) Formulasi perhitungan yang digunakan adalah sebagai berikut:

———————— „„(2)

Pada Tabel 1 dan persamaan (2) TP merupakan *True Positive*, TN merupakan *True Negative*, FP merupakan *False Positive*, dan FN merupakan *False Negative*.

4. HASIL DAN PEMBAHASAN

4.1 *Preprocessing Data*

Langkah *preprocessing* dijelaskan sebagai berikut:

4.1.1 Mengidentifikasi banyaknya *missing values*

Untuk mengidentifikasi banyaknya *missing values* pada setiap fitur dengan sintaks *isna().sum()* yang ada pada library *python 3*. Pada *dataset default*, nilai *missing values* dituliskan dengan angka 0 (nol) pada setiap fitur kelas input, sehingga pada saat dilakukan identifikasi *missing values* menggunakan sintaks *isna().sum()* angka nol tersebut tidak teridentifikasi sebagai nilai *missing values*, karena dianggap mempunyai nilai yakni nol. Oleh karena itu, nilai *missing value* tersebut perlu diubah dulu menjadi kosong (NaN) dengan sintaks *replace* nol menjadi NaN yang merupakan salah satu *library* dari *Numpy*. Tampilan awal identifikasi *missing values* saat masih bernilai nol dapat dilihat pada Tabel 2. Sedangkan tampilan identifikasi nilai *missing values* setelah angka nol sebagai representasi *missing values* diubah menjadi *NaN* disajikan pada Tabel 3.

Tabel 2 Identifikasi *missing values* pada *dataset default*

Fitur	Banyak *missing values*
Kehamilan (Preg)	0
Glukosa (Glu)	0
Tekanan Darah (BP)	0
Ketebalan Kulit (ST)	0
Insulin (Ins)	0
BMI	0
Fungsi Pedigree	0
Diabetes (DP)	0
Usia (Age)	

Tabel 3 Identifikasi *missing values* pada dataset setelah nilai 0 sebagai representasi *missing values* diubah menjadi *NaN*.

Fitur	Banyaknya *missing values*
Kehamilan (Preg)	111
Glukosa (Glu)	5
Tekanan Darah (BP)	35
Ketebalan Kulit (ST)	227
Insulin (Ins)	374
BMI	11
Fungsi Pedigree	0
Diabetes (DP)	0
Usia (Age)	

Adapun tampilan Dataset setelah nol sebagai representasi missing values diubah menjadi NaN disajikan pada Tabel 4.

Tabel 4. Dataset kelas input setelah nilai nol sebagai representasi missing values diubah menjadi NaN

	Preg	Glu	BP	ST	Ins	BMI
0	6.0	148.0	72.0	35.0	NaN	33.6

1	1.0	85.0	66.0	29.0	NaN	26.6
2	8.0	183.0	64.0	NaN	NaN	23.3
...	...	...	...	...	...	...
766	1.0	126.0	60.0	NaN	NaN	30.1
767	1.0	93.0	70.0	31.0	NaN	30.4

	DP	Age
0	0.627	50
1	0.351	31
2	0.672	32
...	...	...
766	0.349	47
767	0.315	23

4.1.2 *Imputing missing values*

Teknik *impute* (Brownley, 2020) merupakan metode pendekatan dalam statistika untuk mengestimasi suatu *missing values* pada suatu fitur dalam dataset, kemudian menggantikan semua *missing values* dengan suatu nilai statistik tertentu. Nilai statistik tertentu tersebut dapat ditentukan dengan mean, median, nilai konstan, serta modus.

Teknik *impute* yang digunakan adalah *impute* menggunakan *mean*, *median* dan *modus* dari setiap fitur. Tampilan dataset setelah nilai missing values diimpute menggunakan nilai mean, median dan modus ditampilkan pada Tabel 5, Tabel 6 dan Tabel 7

Tabel 5. Imputing Missing Values dengan Mean

	Preg	Glu	BP	ST	Ins	BMI
0	6.0	148.0	72.0	35.0	155.5	33.6
1	1.0	85.0	66.0	29.0	155.5	26.6
2	8.0	183.0	64.0	29,15	155.5	23.3
...	...	...	...	...	...	...
766	1.0	126.0	60.0	29,15	155.5	30.1
767	1.0	93.0	70.0	31.0	155.5	30.4

	DP	Age
0	0.627	50
1	0.351	31
2	0.672	32
...	...	...
766	0.349	47
767	0.315	23

Tabel 6. Imputing Missing Values dengan Median

	Preg	Glu	BP	ST	Ins	BMI
0	6.0	148.0	72.0	35.0	125.0	33.6
1	1.0	85.0	66.0	29.0	125.0	26.6
2	8.0	183.0	64.0	29.0	125.0	23.3
...	...	...	...	...	...	...
766	1.0	126.0	60.0	29.0	125.0	30.1
767	1.0	93.0	70.0	31.0	125.0	30.4

	DP	Age
0	0.627	50
1	0.351	31
2	0.672	32
...	...	...
766	0.349	47
767	0.315	23

Tabel 7. Imputing Missing Values dengan Modus

	Preg	Glu	BP	ST	Ins	BMI
0	6.0	148.0	72.0	35.0	NaN	33.6
1	1.0	85.0	66.0	29.0	NaN	26.6
2	8.0	183.0	64.0	NaN	NaN	23.3
...	...	...	...	...	...	...
766	1.0	126.0	60.0	NaN	NaN	30.1
767	1.0	93.0	70.0	31.0	NaN	30.4

	DP	Age
0	0.627	50
1	0.351	31
2	0.672	32
...	...	...
766	0.349	47
767	0.315	23

4.2. Klasifikasi *Adaboost*

4.2.1 Uji coba mengunakan dataset yang mengandung *NaN* sebagai representasi *missing values*

Algoritma *Adaboost* sangat sensitif terhadap keberadaan *missing values*, hal ini dapat dilihat pada saat menjalankan algoritma dengan keberadaan *missing values (NaN)*, terdapat keterangan *ValueError: Input contains NaN*. Adapun eksekusi algoritma pada *python 3* ketika masih terdapat *missing values* terlihat pada notifikasi error berikut ini:

p-ISSN : 2502-5724; e-ISSN : 2541-5735

```
ValueError: Input contains NaN, infinity or a value too large for dtype('float64').
```

Oleh karena itu perlu dilakukan teknik *imputing missing values*. Dalam penelitian ini, akan digunakan 3 teknik *imputing* yakni *imputing* menggunakan nilai *mean*, nilai *median*, dan nilai *modus* dari setiap fitur, kemudian setiap dataset hasil *imputing mean, median, dan modus* diklasifikasikan menggunakan algoritma *Adaboost Classifier* dan dilihat nilai akurasinya

4.2.2 Uji coba mengunakan dataset hasil imputing *mean*

. Hasil klasifikasi algoritma *Adaboost Clasifier* pada dataset hasil *imputing mean* menghasilkan akurasi sebesar 80.09 %.

4.2.3 Uji coba mengunakan dataset hasil imputing *median*

Dataset hasil *imputing median* menghasilkan akurasi sebesar 76.19 %

4.2.4 Uji coba mengunakan dataset hasil imputing *modus*

Dataset hasil *imputing* modus, *output* yang dihasilkan sama dengan dataset yang mengandung *NaN* sebagai representasi *missing values*. Hal ini dikarenakan, aplikasi mengenali *NaN* sebagai nilai modus untuk beberapa fitur, dengan demikian, nilai *NaN* tetap ada.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

1. Hasil klasifikasi algoritma *Adaboost Clasifier* pada dataset hasil *imputing mean* menghasilkan akurasi sebesar 80.09 %
2. Datasethasil*imputingmedian* menghasilkan akurasi sebesar 76.19 %
3. Dataset setelah *imputing* modus, diperoleh hasil yang sama dengan *default dataset* yang belum dilakukan *imputing*, akibatnya *Adaboost classifier* tidak bisa berjalan karena *Adaboost* sangat sensitif terhadap *missing values*. Nilai *missing values* untuk beberapa fitur paling sering muncul, dikenali sebagai modus oleh *python* sehingga nilai *missing values* digantikan dengan *NaN*

5.2 Saran

Penelitian ini dapat dikembangkan lebih lanjut dengan:

1. Menggunakan algoritma klasifikasi lain, khususnya algoritma klasifikasi *boosting*, seperti *XGBoost classifier* , *Gradient Boosting Trees (GBT)*, dan sebagainya.
2. Menggunakan teknik *imputing* lainnya, misalnya: *simpleimputer, kmeans, global most common, concept most common, atau menggunakan beberapa algoritma data mining yang biasa digunakan, seperti KNN, K-means, dan Support Vector Machine (SVM).*
3. Diujikan menggunakan dataset lain, khususnya dataset yang memiliki label *multi class* keputusan, tidak terbatas pada *binary class*.

DAFTAR PUSTAKA

Bimo, P., Setio, N., Retno, D., Saputro, S., & Winarno, B. (2020). Klasifikasi dengan Pohon Keputusan Berbasis Algoritme C4.5. *PRISMA, Prosiding Seminar Nasional Matematika*, *3*, 64–71.

Bisri, A. (2015). Penerapan Adaboost untuk Penyelesaian Ketidakseimbangan Kelas pada Penentuan Kelulusan Mahasiswa dengan Metode Decision Tree. *Journal of Intelligent Systems*, *1*(1), 27–32.

Brownley, J. (2020). *Statistical Imputation for Missing Values in Machine Learning*. https://machinelearningmastery.com/statistical-imputation-for-missing-values-in-machine-learning/

Defiyanti, S. (2015). *Integrasi Metode Klasifikasi Dan Clustering dalam Data Mining*. *March*, 39–44.

Hestiana, D. W. (2017). Journal of Health Education. *Journal of Health Education*, *25*(1), 57–60. https://doi.org/10.1080/10556699.1994.10603001

p-ISSN : 2502-5724; e-ISSN : 2541-5735

Imaduddin, Z., & Tawakal, H. A. (2015). *Deteksi dan klasifikasi daun menggunakan metode adaboost dan svm*. 6–8.

Kurniawati, G. N. (2021). *Algoritma Machine Learning yang Harus Kamu Pelajari di Tahun 2021*. https://www.dqlab.id/algoritma-machine-learning-yang-perlu-dipelajari

Lukman, A., & Marwana. (2014). Machine Learning Multi Klasifikasi Citra Digital. *Konferensi Nasional Ilmu Komputer (KONIK)*, *December 2014*, 1–6.

Mulyati, S., Informatika, T., Pamulang, U., & Pelanggan, C. (2017). *Ketidakseimbangan Kelas Berbasis Naïve Bayes Pada Prediksi*. *2*(4), 190–199.

Oktanisa, I., & Supianto, A. A. (2018). Perbandingan Teknik Klasifikasi Dalam Data Mining Untuk Bank a Comparison of Classification Techniques in Data Mining for. *Teknologi Informasi Dan Ilmu Komputer*, *5*(5), 567–576. https://doi.org/10.25126/jtiik20185958

Purwadhika, S. S. (2019). *Apa Itu Python dan Fungsinya di Dunia Nyata?* https://medium.com/purwadhikaconnect/apa-itu-python-dan-fungsinya-di-dunia-nyata-d5b533117c63

Rohman, A., Suhartono, V., & Supriyanto, C. (2017). Penerapan Agoritma C4.5 Berbasis Adaboost Untuk Prediksi Penyakit Jantung. *Jurnal Teknologi Informasi*, *13*, 13–19.

HASIL_60150823

ORIGINALITY REPORT

2%	1%	1%
SIMILARITY INDEX	INTERNET SOURCES	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Middlesex University Student Paper	1%
2	projets.lam.fr Internet Source	1%

Exclude quotes	On	Exclude matches	On
Exclude bibliography	On		