Gunung Djati Conference Series, Volume 3 (2021)
Mini Seminar Kelas *Data Mining* 2020
ISSN: 2774-6585
Website: https://conferences.uinsgd.ac.id/gdcs

Klasifikasi Diagnosa Penyakit Jantung menggunakan Algoritma Random Forest

(*Classification of Heart Disease Diagnosis using the Random Forest Algorithm*)

Akbar Hidayatullah Harahap[1], Ihsan Muttaqin Bin Abdul Malik[2], Muhammad Irfan Nur Imam[3], Muhammad Thariq Sabiq Bilhaq[4], Aisyah Amini Nur[5], Siti Lutfia Dwi Agustini[6]

[1]Teknik Informatika, UIN Sunan Gunung Djati Bandung, 1177050012@student.uinsgd.ac.id
[2]Teknik Informatika, UIN Sunan Gunung Djati Bandung, 1177050053@student.uinsgd.ac.id
[3]Teknik Informatika, UIN Sunan Gunung Djati Bandung, 1177050071@student.uinsgd.ac.id
[4]Teknik Informatika, UIN Sunan Gunung Djati Bandung, 1177050077@student.uinsgd.ac.id
[5]Teknik Informatika, UIN Sunan Gunung Djati Bandung, 1177050010@student.uinsgd.ac.id
[6]Teknik Informatika, UIN Sunan Gunung Djati Bandung, 1177050107@student.uinsgd.ac.id

Abstrak

Jantung merupakan salah satu bagian penting dalam organ tubuh manusia. Namun tidak menutup kemungkinan jantung ini bermasalah dan menimbulkan beberapa gejala, sehingga menimbulkan penyakit Jantung yang mematikan. Oleh karena itu banyak penelitian digunakan untuk memperoleh data Diagnosa Penyakit Jantung yang cepat, tepat dan akurat. Salah satunya yaitu dengan klasifikasi diagnosa penyakit jantung menggunakan algoritma Random Forest. Metode algoritma random forest ini menggunakan beberapa pohon keputusan yang disatukan. Sehingga dapat diperoleh hasil yang akurat mengenai diagnosa penyakit jantung ini. Akurasi yang diperoleh dari hasil eksperimen menggunakan Bahasa pemrograman Python ini adalah 85,3%

Kata kunci: data mining, diagnosa, klasifikasi, random forest

Abstract

The heart is an important part of the human body organs. But it does not rule out this heart problem and causes several symptoms, causing deadly heart disease. Therefore, many studies are used to obtain fast, precise and accurate heart disease diagnosis data. One of them is the classification of heart disease diagnoses using the Random Forest algorithm. This random forest algorithm method uses several unified decision trees. So that accurate results can be obtained regarding the diagnosis of this heart disease. The accuracy obtained from the experimental results using the Python programming language is 85.3%.

Keywords: *classification, data mining, diagnose, random forest*

1 Pendahuluan

Menurut Organisasi Kesehatan Dunia (WHO), penyakit jantung adalah penyebab kematian pertama di negara berpenghasilan tinggi dan rendah dan terjadi hampir sama pada pria dan wanita. Tidak hanya berdampak besar pada individu dan kualitas hidup mereka secara umum, tetapi juga pada biaya kesehatan masyarakat dan perekonomian negara. Diagnosis penyakit jantung merupakan keputusan yang lebih mahal dalam diagnosis. Banyak Artificial Intelligence (AI) digunakan dalam diagnosis medis. Seiring perkembangan teknologi informasi, menambah akumulasi data di berbagai bidang

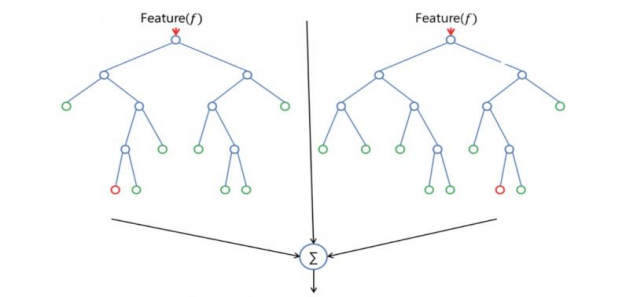
Gunung Djati Converence Series
ISSN 2774-6585
Volume 3 (2021)

telah meningkat sehingga banyak informasi dikenal. Data yang terkumpul dapat dianalisis hingga dapat mengungkapkan informasi berguna yang tersembunyi [1]. Dengan melakukan penambangan data, yang merupakan ilmu baru, kami dapat mengekstrak pengetahuan tersembunyi dari data tersebut. Fungsi inti dari data mining menerapkan berbagai metode dan algoritma untuk menemukan data yang disimpan. Melakukan penambangan data mengungkapkan adanya hubungan yang berguna di antara data, dan aturan ini dapat diterapkan untuk pengambilan keputusan yang tepat [2], [3]. Tujuan dari penelitian ini adalah melakukan proses klasifikasi yang dapat digunakan dalam mendiagnosa penyakit jantung. Terdapat beberapa penelitian terdahulu yang terkait penelitian ini, antara lain: (1) penerapan algoritma Random Forest untuk diagnosis penyakit diabetes [4]; (2) diagnosis penyakit ginjal kronis menggunakan algoritma Random Forest [5]; diagnosis penyakit kangker servik menggunakan algoritma Random Forest; dan [6] diagnosis penyakit tanaman tomat dengan menggunakan algoritma Random Forest [7].

2 Metodologi

2.1 Random Forest Classifier

Random forest classifier terdiri dari kombinasi pengklasifikasi pohon di mana setiap pengklasifikasi dihasilkan menggunakan vektor acak yang diambil sampelnya secara independen dari vektor masukan, dan setiap pohon memberikan suara unit untuk kelas paling populer untuk mengklasifikasikan masukan vector [8]. Pengklasifikasi hutan acak yang digunakan untuk penelitian ini terdiri dari menggunakan fitur yang dipilih secara acak atau kombinasi fitur di setiap node. Random forest classifier (RFC) adalah salah satu teknik pembelajaran ensemble paling sukses yang telah terbukti menjadi teknik yang sangat populer dan kuat dalam pengenalan pola dan pembelajaran mesin untuk klasifikasi dimensi tinggi dan masalah miring [8]. Kelemahan yang terkait dengan pengklasifikasi pohon adalah variansnya yang tinggi. Dalam praktiknya, tidak jarang perubahan kecil dalam kumpulan data pelatihan menghasilkan pohon yang sangat berbeda. Alasannya terletak pada sifat hierarki dari pengklasifikasi pohon. Kesalahan yang terjadi pada simpul yang dekat dengan akar pohon menyebar hingga ke daun. Untuk membuat klasifikasi pohon lebih stabil, metodologi decision forest telah ditemukan [9]. Sesuai dengan namanya, metoe ini menciptakan hutan (forest) dengan sejumlah pohon (tree). Secara umum, semakin banyak pohon pada sebuah huhtan maka semakin kuat juga hutan tersebut telihat. Pada kasus yang sama, semakin banyak tree, maka semakin besar pula akurasi yang didapatkan [10]. Satu keuntungan besar dari random forest adalah dapat digunakan untuk masalah klasifikasi dan regresi, yang merupakan sebagian besar sistem pembelajaran mesin saat ini. Mari kita lihat forest acak dalam klasifikasi, karena klasifikasi terkadang dianggap sebagai elemen penyusun machine learning. Di bawah ini bagaimana hutan acak akan terlihat dengan dua pohon:



Bagan 1 Tree Random Forest

2.2 Data Penyakit Jantung

Data yang digunakan pada penelitian ini adalah Hungarian Institute of Cardiology. Kami menggunakan subset 14 atribut. Untuk data penyakit jantung tersedia di https://www.kaggle.com/ronitf/heart-disease-uci.

Tabel 1 Data Penyakit jantung

Name	Type	Description
Age	Continuous	Age in years
Sex	Discrete	1 = male
		0 = female
Cp	Discrete	Chest pain type:
		1 = typical angina
		2 = atypical angina
		3 = non-angina pa
		4 =asymptomatic
Trestbps	Continuous	Resting blood pressure (in mm Hg)
Chol	Continuous	Serum cholesterol in mg/dl
Fbs	Discrete	Fasting blood sugar > 120 mg/dl:
		1 = true
		0 = false
Restecg	Discrete	Resting electrocardiographic results:
		0 = normal
		1 = having ST-T wave abnormality
		2 =showing probable or define left ventricular hypertrophy by
		Estes'criteria
Thalach	Continuous	Maximum heart rate achieved
Exang	Discrete	Exercise induced angina:
		1 = yes
		0 = no
Slope	Discrete	The slope of the peak exercise segment :
		1 = up sloping
		2 = flat
		3= down sloping
Diagnosis	Discrete	Diagnosis classes:
		0 = healthy
		1= possible heart disease

3 Hasil dan Pembahasan

3.1 Input Data

Random Forests Algorithm (RFA) merupakan kombinasi dari presdiksi pohon (*tree prediction*) yang setiap pohonnya bergantung pada nilai dari sampel vektor yang acak dan memiliki distribusi yang untuk semua pohon (*tree*) di dalam hutan (*forest*). Algoritma ini akan membuat banyak *Decision Tree* yang masing-masing dari mereka akan memilih (*vote*) kelas atau kesimpulan yang paling tepat. *Random Forest* akan memilih kelas atau kesimpulan yang paling banyak sebagai keputusan akhir [11].

Gunung Djati Conference Series
ISSN 2774-6585
Volume 3 (2021)

Sebelum mengimplementasikan algoritma ini, kita perlu untuk memanggil *libray* yang akan digunakan dan meng-*input Dataset* yang akan diolah. Pada artikel ini *dataset*-nya mengenai penyakit jantung (*hearts disease ICU*), dengan total 296 baris dan 14 kolom, dan *library* RFA yang digunakan adalah **RandomForestClassifier** dari **sklearn.ensemble**.

```
from sklearn.ensemble import RandomForestClassifier
import numpy as np
import pandas as pd
```

```
[ ] data_url = 'https://raw.githubusercontent.com/Qhiba/heart-disease-clasification/master/Heart%20Disease%20UCI.csv'
    dataset = pd.read_csv(data_url)
```

```
[ ] dataset.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

Gambar 1 Input Dataset

Penjelasan dari nama kolom yang terdapat pada dataset adalah sebagai berikut:

1. ***age***: umur dari pasien;
2. ***sex***: jenis kelamin pasien;
    a. Nilai 0 = perempuan;
    b. Nilai 1 = laki-laki.
3. **cp**: *chest pain* atau nyeri dada yang dirasakan;
    a. Nilai 0: *asymptomatic*;
    b. Nilai 1: *atypical angina*;
    c. Nilai 2: *non-anginal pain*;
    d. Nilai 3: *typical angina*.
4. **trestbps**: tekanan darah pasien saat istirahat / *resting blood preasure* (satuan mm Hg saat pasien di rawat inap);
5. **chol**: tingkat kolesterol pasien dalam mg/dl;
6. **fbs**: tingkat gula darah pasien setela melakukan *Fasting Blood Sugar* (FBS). Nilai menyatakan tingkat gula darah lebih tinggi dari 120 mg/dl atau tidak;
    a. Nilai 0 = tidak;
    b. Nilai 1 = Iya.
7. **restecg**: *resting electrocardiographic measurement* hasil dari hitungan *electrocardiographic* ketika pasien beristirahat;
    a. Nilai 0: kemungkinan terjadi *left ventricular hypertrophy* (LVH) dimana pompa jatung bagian kiri membengkak dan tidak memompa secara efisien;
    b. Nilai 1: normal;
    c. Nilai 2: memiliki kelainan pada ST-T wave.
8. **thalach**: tingkat tertinggi detak jantung ketika sedang melakukan stressing (contohnya ketika sedang olahraga);
9. **exang**: apakah pasien mengindap angina ketika berolahraga;
    a. Nilai 0: Tidak;

   b. Nilai 1: Iya.
10. **oldpeak**: Menurunnya ST *segment* saat melakukan olahraga relatif dengan pada saat beristirahat;
11. **slope**: lembah pada ST *segment* pada bagian puncak ketika berolahraga;
    a. Nilai 0: descending;
    b. Nilai 1: flat;
    c. Nilai 2: ascending;
12. **ca**: Jumlah pembuluh darah utama yang diwarnai oleh pewarna radioaktif (nilai 0 sampai 3);
13. **thal**: merupakan hasil observasi pada aliran darah via pewarna radioaktif;
    a. Nilai 1: *fixed defect* atau cacat total (tidak ada aliran darah pada sebagian daerah jantung);
    b. Nilai 2: *normal blood flow* atau aliran darah normal;
    c. Nilai 3: *reversible defect* (hasil observasi menunjukan aliran darah tidak normal).
14. **target**: target variabel yang menunjukkan apakah pasien menderita penyakit jantung atau tidak;
    a. Nilai 0: Iya / penyakit jantung / *heart disease*;
    b. Nilai 1: Tidak / Sehat / *Healthy*.

Selanjutnya mengubah nama dari semua kolom pada table dan nama dari target agar lebih mudah dimengerti dengan memasukan script berikut:

```
[102] dataset.columns = ['age', 'sex', 'chest_pain_type', 'resting_blood_pressure', 'cholesterol',
                         'fasting_blood_sugar', 'rest_ecg', 'max_heart_rate_achieved', 'exercise_induced_angina',
                         'st_depression', 'st_slope', 'num_major_vessels', 'thalassemia', 'target']

      dataset['target'][dataset['target'] == 0] = 'Heart Disease'
      dataset['target'][dataset['target'] == 1] = 'Healthy'

      dataset
```

de/indexing.html#returning-a-view-versus-a-copy

rest_ecg	max_heart_rate_achieved	exercise_induced_angina	st_depression	st_slope	num_major_vessels	thalassemia	target
0	150	0	2.3	0	0	1	Healthy
1	187	0	3.5	0	0	2	Healthy
0	172	0	1.4	2	0	2	Healthy
1	178	0	0.8	2	0	2	Healthy
1	163	1	0.6	2	0	2	Healthy
...	...	...	...	...	...	...	...
1	123	1	0.2	1	0	3	Heart Disease
1	132	0	1.2	1	0	3	Heart Disease
1	141	0	3.4	1	2	3	Heart Disease
1	115	1	1.2	1	1	3	Heart Disease
0	174	0	0.0	1	1	2	Heart Disease

Gambar 2 Mengubah Nama pada Kolom dan Nama Target

Gunung Djati Converence Series
ISSN 2774-6585
Volume 3 (2021)

3.2 Pre-Processing Data

Pada tahap ini *pre-processing* ini terdapat beberapa hal yang perlu dilakukan, yaitu:

1. Membagi dataset menjadi dua bagian, yaitu (1) *training data*, merupakan data yang akan dilatih oleh RFA, dan (2) *test data*, merupakan data yang akan diklasifikasi oleh RFA.

2. Membuat dua dataset baru, yaitu (1) *train*, yang berisikan *training data*, dan (2) test, yang berisikan *test*.

Proses pembagian dataset menjadi dua bagian adalah dengan menambahkan kolom baru yang berisikan apakah suatu barus akan di latih atau tidak. Persentasi pembagian dataset pada artikel ini adalah data <= 75% untuk data yang akan dilatih dan sisanya untuk data yang akan diklasifikasikan.

```
[104] # Membagi data menjadi data yang akan dilatih (Train Data) dan data yang akan di test (Test Data).
      dataset['is_train'] = np.random.uniform(0, 1, len(dataset)) <= .75

      # Tampilkan data
      dataset
```

ax_heart_rate_achieved	exercise_induced_angina	st_depression	st_slope	num_major_vessels	thalassemia	target	is_train
150	0	2.3	0	0	1	Healthy	False
187	0	3.5	0	0	2	Healthy	True
172	0	1.4	2	0	2	Healthy	False
178	0	0.8	2	0	2	Healthy	True
163	1	0.6	2	0	2	Healthy	True
...	...	...	...	...	...	...	...
123	1	0.2	1	0	3	Heart Disease	True
132	0	1.2	1	0	3	Heart Disease	True
141	0	3.4	1	2	3	Heart Disease	True
115	1	1.2	1	1	3	Heart Disease	True
174	0	0.0	1	1	2	Heart Disease	False

Gambar 3 Membagi Dataset menjadi training data dan test data

Setelah dataset selesai dibagi, dibuatlah dataset baru untuk menyimpan *training data* dan *test data*.

```
[105] # Membuat dataframe dengan baris tes (test rows) dan baris latihan (training rows)
      train, test = dataset[dataset['is_train']==True], dataset[dataset['is_train']==False]

      # Munculkan jumlah dari observasi untuk dataframe test dan training
      print('Jumlah observasi dalam training data', len(train))
      print('Jumlah observasi dalam test data', len(test))

      Jumlah observasi dalam training data 221
      Jumlah observasi dalam test data 75
```

Gambar 4 Membuat Data Set baru dan menghitung total data yang ada didalamnya

Dalam percobaan ini, jumlah data yang akan dilatih sebanyak 221 data dan jumlah data yang akan di tes atau klasifikasikan sebanyak 75 data.

3.3 Implementasi Algoritma

Setelah proses *pre-processing* selesai, selanjutnya adalah mengimplementasikan RFA. Diawali dengan membuat *Random Forest Classifier* yang akan digunakan untuk melatih dan mengklasifikasikan data. Setelah itu latihlah *train dataset* dengan membandingkan seluruh kolom yang ada dengan kolom 'target'.

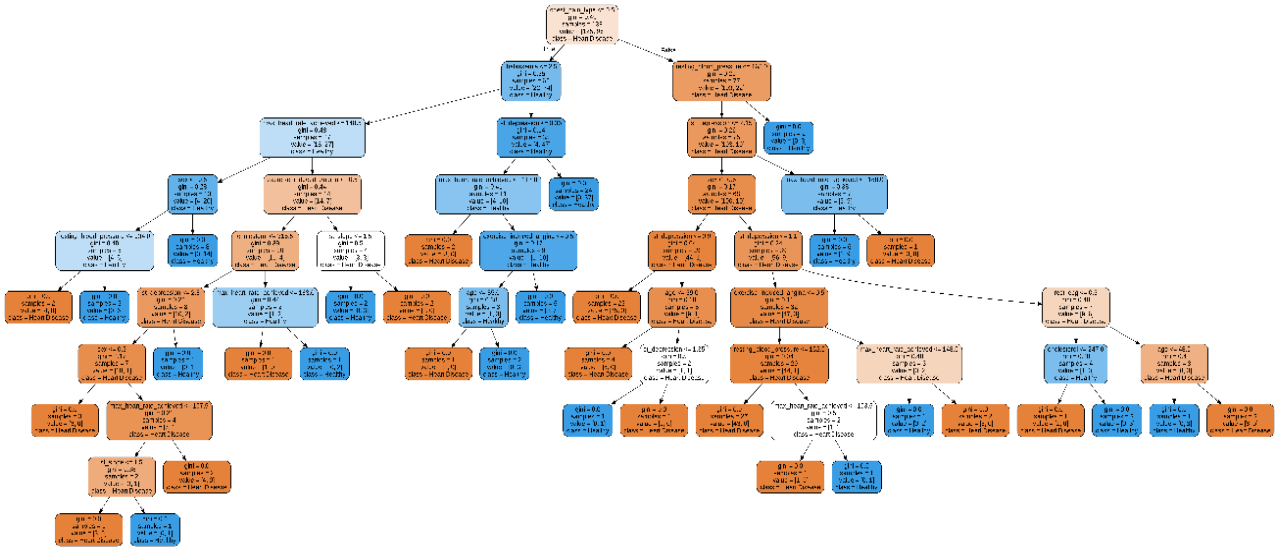
```
[107] # Membuat Random Forest Classifier.
      clf = RandomForestClassifier(n_jobs=2, random_state=0)

      # Melatih Classifier
      rfa = clf.fit(train[dataset.columns[:13]], train['target'])
      rfa

      RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                             criterion='gini', max_depth=None, max_features='auto',
                             max_leaf_nodes=None, max_samples=None,
                             min_impurity_decrease=0.0, min_impurity_split=None,
                             min_samples_leaf=1, min_samples_split=2,
                             min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=2,
                             oob_score=False, random_state=0, verbose=0,
                             warm_start=False)
```

Gambar 5 Membuat Random Forest Classifier dan melatih data

Pada *decision tree*, langkah awal untuk membuat *tree* adalah menentukan kolom mana yang akang menjadi *root* dengan membandingkan setiap kolom dengan kolom 'target'. Setelah itu dihitunglah *gini impurity* dari masing-masing perbandingan, dan memilih kolom yang memiliki *gini impurity* yang paling kecil sebagai *root*. Penentuan *nodes* pada bagian kiri dan kanan hingga penentuan *leaves* pun ditentukan dengan cara yang sama, bedanya adalah dari jumlah data yang digunakan untuk perhitunga *gini impurity* dan kolom yang menjadi *root* atau *parent* dari *nodes* tidak akan diikut sertakan.

Cara yang sama dilakukan oleh RFA pada saat membuat *tree*, hanya bedanya RFA akan terlebih dulu membuat table *bootstrap* yang isinya diambil dari *training data* yang barisnya diacak. *Tree* akan dibuat mengacu kepada tabel *bootstrap* yang sudah dibuat sehingga jumlah nya akan mengikuti jumlah dari tabel *bootstrap* yang ada. Jumlah tree yang akan dibuat ditentukan oleh parameter bernama **n_estimators**. Pada percobaan ini jumlah *tree* yang dibuat oleh RFA mengikutti *default* dari *library* 100 *tree*. Berikut adalah salah satu *tree* yang selesai dibuat.

Gunung Djati Conference Series
ISSN 2774-6585
Volume 3 (2021)



Gambar 6 Salah satu tree yang ada dalam Random Forest

Setelah selesai melakukan pelatihan data, saatnya klasifikasi pada *test data* dilakukan. RFA akan memprediksi nilai '*target*' dari masing-masing baris yang ada di *test data* dengan memasukan dan membandingkan nilai yang ada kolom dengan seluruh *tree* yang sudah dibuat sebelumnya dan hasil akhir keputusannya adalah nilai yang paling banyak muncul. Berikut adalah hasil prediksinya.

```
[110] # Mengaplikasikan Classifier yang sudah dilatih kedalam test data
      predicted_target = clf.predict(test[dataset.columns[:13]])
      predicted_target

      array(['Healthy', 'Healthy', 'Healthy', 'Healthy', 'Healthy', 'Healthy',
             'Healthy', 'Healthy', 'Healthy', 'Heart Disease', 'Healthy',
             'Healthy', 'Healthy', 'Healthy', 'Healthy', 'Healthy', 'Healthy',
             'Healthy', 'Healthy', 'Healthy', 'Healthy', 'Healthy', 'Healthy',
             'Healthy', 'Healthy', 'Heart Disease', 'Healthy', 'Healthy',
             'Healthy', 'Healthy', 'Healthy', 'Healthy', 'Healthy', 'Healthy',
             'Healthy', 'Healthy', 'Healthy', 'Healthy', 'Heart Disease',
             'Heart Disease', 'Healthy', 'Heart Disease', 'Heart Disease',
             'Healthy', 'Heart Disease', 'Heart Disease', 'Heart Disease',
             'Heart Disease', 'Heart Disease', 'Heart Disease', 'Heart Disease',
             'Heart Disease', 'Heart Disease', 'Heart Disease', 'Healthy',
             'Heart Disease', 'Heart Disease', 'Heart Disease', 'Heart Disease',
             'Heart Disease', 'Heart Disease', 'Healthy', 'Heart Disease',
             'Heart Disease', 'Healthy', 'Heart Disease', 'Heart Disease',
             'Healthy', 'Heart Disease', 'Heart Disease', 'Heart Disease',
             'Healthy', 'Heart Disease', 'Heart Disease', 'Healthy'],
            dtype=object)
```

Gambar 7 Hasil Prediksi RFA dari test data

Kita dapat menampilkan probabilitas yang diprediksi pada 10 observasi baris pertama.

```
[111] # Menampilkan probabilitas yang diprediksi pada 10 observasi pertama
      clf.predict_proba(test[dataset.columns[:13]])[0:10]

      array([[0.59, 0.41],
             [0.97, 0.03],
             [0.74, 0.26],
             [0.84, 0.16],
             [0.82, 0.18],
             [0.95, 0.05],
             [0.91, 0.09],
             [0.69, 0.31],
             [0.8 , 0.2 ],
             [0.48, 0.52]])
```

Gambar 8 Probabilitas yang diprediksi pada 10 Baris pertama

Menghitung keakuratan dari RFA dalam memprediksi nilai '*target'* pada *test data* dengan membandingkan menghitung selisih antara nilai 'target' pada data asli dengan nilai 'target' hasil prediksi. Selisih dapat lebih mudah dilihat dengan membuat *confusion matrix*.

Predicted Target	Healthy	Heart Disease
Actual Target		
Healthy	36	3
Heart Disease	8	28

Gambar 9 Confusing Matriks

Dapat kita lihat bahwa hasil prediksi RFA yang sesuai dengan data asli sebanyak 64 data dan yang tidak sesuai sebanyak 11 data. Nilai keakuratan dapat dihitung dengan menghitung **mean** data prediksi yang tepat dikalikan 100.

$$accuracy = \frac{jumlah\ data\ yang\ tepat}{total\ data}\ x\ 100 \quad (1)$$

$$accuracy = \frac{64}{75}\ x\ 100 = 85.3\%$$

Sehingga keakuratan RFA untuk memprediksi data penyakit jantung ini sebesar 85.3%.

4 Simpulan

Berdasarkan hasil pengujian Algoritma Random Forest, dapat diambil beberapa kesimpulan bahwa Algoritma Random Forest berhasil di lakukan dengan data diagnosa penyakit Jantung dengan data yang tersedia. Algoritma Random Forest memiliki keberhasilan tinggi untuk mengklasifikasikan sebuah kasus klasifikasi penyakit jantung. Kelebihan dari Algoritma Random forest dapat mengatasi noise dan missing value dan dapat mengatasi data dalam jumlah besar. Walaupun kekurangan dari Algoritma Random Forest adalah interpretasi yang sulit serta membutuhkan tuning model yang tepat untuk datanya.

Gunung Djati Converence Series
ISSN 2774-6585
Volume 3 (2021)

Referensi

[1] H. Bagheri and A. A. Shaltooki, "Big data: Challenges, opportunities and cloud based solutions," *Int. J. Electr. Comput. Eng.*, vol. 5, no. 2, pp. 340–343, 2015, doi: 10.11591/ijece.v5i2.pp340-343.

[2] V. Paramasivam, T. S. Yee, S. K. Dhillon, and A. S. Sidhu, "A methodological review of data mining techniques in predictive medicine: An application in hemodynamic prediction for abdominal aortic aneurysm disease," *Biocybernetics and Biomedical Engineering*, vol. 34, no. 3. pp. 139–145, 2014, doi: 10.1016/j.bbe.2014.03.003.

[3] K. C. Tan, E. J. Teoh, Q. Yu, and K. C. Goh, "A hybrid evolutionary algorithm for attribute selection in data mining," *Expert Syst. Appl.*, vol. 36, no. 4, pp. 8616–8630, 2009, doi: 10.1016/j.eswa.2008.10.013.

[4] S. Benbelkacem and B. Atmani, "Random forests for diabetes diagnosis," in *2019 International Conference on Computer and Information Sciences, ICCIS 2019*, 2019, doi: 10.1109/ICCISci.2019.8716405.

[5] A. Subas, E. Alickovic, and J. Kevric, "Diagnosis of chronic kidney disease by using random forest," in *IFMBE Proceedings*, 2017, vol. 62, pp. 589–594, doi: 10.1007/978-981-10-4166-2_89.

[6] G. Sun, S. Li, Y. Cao, and F. Lang, "Cervical cancer diagnosis based on random forest," *Int. J. Performability Eng.*, vol. 13, no. 4, pp. 446–457, 2017, doi: 10.23940/ijpe.17.04.p12.446457.

[7] M. Govardhan and M. B. Veena, "Diagnosis of Tomato Plant Diseases using Random Forest," in *2019 Global Conference for Advancement in Technology, GCAT 2019*, 2019, doi: 10.1109/GCAT47503.2019.8978431.

[8] L. Breiman, "Random forests," *UC Berkeley TR567*, 1999.

[9] T. K. Ho, "Random decision forests," in *Proceedings of 3rd international conference on document analysis and recognition*, 1995, vol. 1, pp. 278–282.

[10] S. Polamuri, "How the random forest algorithm works in machine learning," *Retrieved December*, vol. 21, 2017.

[11] Y. L. Pavlov, *Random forests*. Walter de Gruyter GmbH & Co KG, 2019.