

Sistem Prediksi Penyakit Jantung Menggunakan Metode *Naive Bayes*

Dimsyar M Al Hafiz¹, Khoirul Amaly¹, Javen Jonathan¹, M. Teranggono Rachmatullah¹, Rosidi¹

¹Teknik Elektro, Fakultas Teknik

Universitas Sriwijaya

Palembang, Indonesia

Penulis korespondensi: khoirulamaly@gmail.com

Abstrak— Penyakit jantung adalah salah satu penyakit yang banyak memakan korban jiwa. Penyakit ini mengancam semua usia. Berbagai faktor dapat menjadi tolak ukur seseorang mengidap penyakit jantung. *Machine learning* adalah subdivisi yang muncul dari kecerdasan buatan. Fokus utamanya adalah merancang sistem, memungkinkan mereka untuk belajar dan membuat prediksi berdasarkan pengalaman. Ini melatih algoritma pembelajaran mesin menggunakan kumpulan data pelatihan untuk membuat model. Model menggunakan data *input* baru untuk memprediksi penyakit jantung. Faktor-faktor tersebut dapat diklasifikasikan menggunakan metode *Naive Bayes*. Eksperimen simulasi dilakukan untuk melihat apakah sistem yang dibangun menggunakan metode *Naive Bayes* mampu untuk memprediksi apakah seseorang mengidap penyakit jantung. Berdasarkan eksperimen data uji berjumlah 61 data, 52 diantaranya terprediksi benar dengan nilai keakuratan sebesar 85,25%. Ke depannya diperlukan sistem prediksi penyakit jantung menggunakan metode klasifikasi lain agar dapat ditemukan metode yang paling akurat dan efektif.

Kata Kunci— *penyakit jantung, machine learning, naive bayes*

Abstract— Heart disease is one of the most common diseases. This disease threatens all ages. Various factors can be a benchmark for someone suffering from heart disease. Machine learning is an emerging subdivision of artificial intelligence. The main focus is on designing systems, enabling them to learn and make predictions based on experience. It trains machine learning algorithms using training data sets to build models. The model uses new input data to predict heart disease. These factors can be classified using the Naive Bayes method. Simulation experiments were conducted to see whether the system built using the Naive Bayes method was able to predict whether a person has heart disease. Based on the experimental test data, there are 61 data, 52 of which are predicted to be correct with an accuracy value of 85.25%. In the future, a heart disease prediction system using other classification methods is needed in order to find which method is the most accurate and effective.

Keywords— *heart disease, machine learning, naive bayes*

I. PENDAHULUAN

Berdasarkan data Badan Kesehatan Dunia (WHO) pada 2016, total kematian di Indonesia mencapai 1.863.000 jiwa. Sebanyak 73 persen di antaranya disebabkan oleh penyakit tidak menular. Adapun jantung koroner menjadi penyebab kematian nomor 1 sebanyak 35 persen. Prevalensi penyakit jantung tertinggi didominasi oleh manula berusia 75 tahun ke atas yakni 4,7 persen. Namun bukan berarti usia muda bebas dari ancaman penyakit jantung. Melihat data Riskesdas 2018, angka prevalensi penyakit jantung di Indonesia sudah dimulai sejak umur kurang dari 1 tahun dan tersebar di semua kelompok umur. Semakin tinggi usia seseorang, maka risiko terkena penyakit jantung akan semakin tinggi [1]. Banyak faktor predisposisi seperti kebiasaan pribadi serta predisposisi genetik menyebabkan penyakit jantung. Berbagai faktor risiko kebiasaan seperti merokok, terlalu sering menggunakan alkohol dan kafein, stres, dan aktivitas fisik bersama dengan faktor fisiologis lainnya seperti obesitas, hipertensi, kolesterol, darah tinggi dan juga kondisi jantung yang sudah ada sebelumnya merupakan faktor predisposisi penyakit jantung. Diagnosis medis penyakit jantung yang efisien dan akurat serta dini memainkan peran penting dalam mengambil tindakan pencegahan untuk mencegah kematian.

Data *mining* adalah proses nontrivial untuk mengidentifikasi valid, baru, berpotensi berguna dan akhirnya pola yang dapat dipahami dalam data dengan penggunaan database yang luas dan pertumbuhan eksplisif dalam ukurannya [2]. Data *mining* mengacu pada ekstraksi informasi yang diperlukan dari kumpulan data besar di berbagai bidang seperti bidang medis, bidang bisnis, dan bidang pendidikan. Algoritma ini dapat menganalisis data yang sangat besar dari berbagai bidang, salah satu bidang penting tersebut adalah bidang medis. Ini adalah pengganti pendekatan pemodelan prediksi rutin menggunakan komputer untuk mendapatkan pemahaman tentang interaksi yang kompleks dan non-linear antara faktor-faktor yang berbeda dengan mengurangi kesalahan dalam hasil prediksi dan faktual. Data *mining* mengeksplorasi kumpulan data besar untuk mengekstrak informasi pengambilan keputusan penting yang tersembunyi dari kumpulan repositori masa lalu untuk analisis di masa mendatang. Bidang medis terdiri dari data pasien yang luar biasa. Data ini perlu diklasifikasikan oleh berbagai algoritma pembelajaran mesin untuk memprediksi

penyakit jantung pada pasien. Berbagai teknik data *mining* seperti regresi, pengelompokan, aturan asosiasi dan teknik klasifikasi seperti *Naive Bayes* yang digunakan sebagai penelitian pada *paper* ini untuk mengklasifikasikan berbagai atribut penyakit jantung dalam memprediksi penyakit jantung.

II. STUDI PUSTAKA

A. Machine Learning

Machine learning adalah subdivisi yang muncul dari kecerdasan buatan. Fokus utamanya adalah merancang sistem, memungkinkan mereka untuk belajar dan membuat prediksi berdasarkan pengalaman. Melatih algoritma pembelajaran mesin menggunakan kumpulan data pelatihan untuk membuat model [3]. Model menggunakan data *input* baru untuk memprediksi penyakit jantung yang menggunakan pembelajaran mesin, yang mendeteksi pola tersembunyi dalam set data *input* untuk membangun model serta membuat prediksi yang akurat untuk kumpulan data baru. *Dataset* dibersihkan dan nilai-nilai yang hilang diisi. Model menggunakan data *input* baru untuk memprediksi penyakit jantung dan kemudian diuji akurasi.

B. Pengklasifikasi Naive Bayes

Pengklasifikasi *Naive Bayes* adalah pengklasifikasi probabilistik berdasarkan teorema *Bayes*, yang mengasumsikan bahwa setiap fitur memberikan kontribusi yang independen dan setara ke kelas target. Mengasumsikan bahwa setiap fitur adalah independen dan tidak berinteraksi satu sama lain, sehingga setiap fitur secara independen dan sama-sama berkontribusi pada kemungkinan sampel untuk menjadi bagian dari kelas tertentu [4]. Pengklasifikasi *Naive Bayes* mudah diimplementasikan dan secara komputasi cepat dan berkinerja baik pada kumpulan data besar yang memiliki dimensi tinggi. Kondusif untuk aplikasi waktu nyata dan tidak sensitif terhadap *noise*. Pengklasifikasi *Naive Bayes* memproses set data pelatihan untuk menghitung probabilitas kelas dan probabilitas bersyarat, yang menentukan frekuensi setiap nilai fitur untuk nilai kelas tertentu dibagi dengan frekuensi *instance* dengan nilai kelas tersebut. Pengklasifikasi *Naive Bayes* berkinerja terbaik ketika fitur yang berkorelasi dihapus karena fitur yang berkorelasi akan dipilih dua kali dalam model yang mengarah ke penekanan berlebihan pada pentingnya fitur yang berkorelasi [5].

Pada persamaan (1) dan persamaan (2), teorema *Bayes* menyediakan cara menghitung posterior probabilitas, $P(c|x)$, dari $P(c)$, $P(x)$, dan $P(x|c)$. Pengklasifikasi *Naive Bayes* mengasumsikan bahwa efek dari nilai a prediktor (x) pada kelas tertentu (c) tidak tergantung pada nilai prediktor lainnya. Asumsi ini disebut kelas kemerdekaan bersyarat [6].

$$P(c | x) = \frac{P(x|c)P(c)}{P(x)} \quad (1)$$

$$P(c | X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c) \quad (2)$$

- $P(c|x)$ adalah probabilitas posterior kelas (target) diberikan prediktor (atribut).
- $P(c)$ adalah probabilitas kelas sebelumnya.
- $P(x|c)$ adalah peluang yang merupakan peluang kelas yang diberikan oleh prediktor.
- $P(x)$ adalah probabilitas sebelumnya dari *predictor*.

III. METODOLOGI PENELITIAN

A. Metode Penelitian

Penelitian ini menggunakan metode eksperimen. Dimana peneliti melakukan percobaan menggunakan dataset yang disediakan oleh *UCI Machine learning*. Dataset ini digunakan sebagai data *training* dengan menggunakan pemrograman bahasa Python dan menggunakan *library* Sklearn untuk pengklasifikasian target berupa diagnosis ada atau tidaknya penyakit jantung menggunakan metode *Naive Bayes*. Peneliti menguji keakuratan dari sistem prediksi yang dibangun dengan menghitung persentase benar atau salah dari hasil prediksi sistem.

B. Pengumpulan Dataset

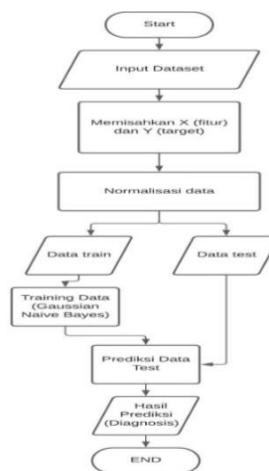
Dataset yang digunakan adalah *Heart Disease* dari publik yang sudah tersebar di internet. Menggunakan dataset untuk melakukan proses *testing* dan juga *training* data yang terdiri dari dataset nyata 304 contoh data dengan 13 fitur, deskripsi, dan *values* yang ditampilkan pada tabel 1. Untuk pengambilan dataset ini dapat kami akses dengan menggunakan internet melalui halaman *website*:<http://archive.ics.uci.edu/ml/datasets/Heart+Disease> [7].

TABEL I. FITUR DATASET

Sr. no	Attribute	Description	Values
1	Age	Age in years	Continuous
2	Sex	Male or female	1 = male

			0 = female
3	Cp	Chest pain type	1 = typical type 1 2 = typical type agina 3 = non-agina pain 4=asymptomatic
4	threstbps	Resting blood pressure	Continuous value in mm hg
5	chol	Serum cholesterol	Continuous value in mm/dl
6	Restecg	Resting electrographic results	0 = normal 1 = having_ ST_T wave abnormal 2 = left ventricular hypertrophy
7	FBS	Fasting blood sugar	1 ≥ 120 mg/dl 0 ≤ 120 mg/dl
8	thalach	Maximum heart rate achieved	Continuous value
9	exang	Exercise-induced agina	0= no 1 = yes
10	oldpeak	ST depression induced by exercise relative to rest	Continuous value
11	slope	The slope of the peak exercise ST segment	1 = unsloping 2 = flat 3 = downsloping
12	Ca	Number of major vessels colored by floursopy	0-3 value
13	thal	Defect type	3 = normal 6 = fixed 7 = reversible defect

C. Rancangan Algoritma Sistem



Gambar 1. Flowchart sistem prediksi penyakit jantung

Pada penelitian ini kami merancang skema sistem prediksi seperti pada Gambar 1. Penelitian ini memiliki tujuan untuk menghasilkan produk berupa sistem prediksi penyakit jantung, sehingga dapat membantu untuk mendiagnosa pasien. Berdasarkan Gambar 1, *flowchart* dimulai dengan *input* dataset yang selanjutnya akan dilakukan pemisahan fitur (X), dan target (Y). Kemudian dilakukan normalisasi data pada fitur dan target tersebut dan mendapatkan hasil data *training* dan juga data test. Pada data *training* ini akan dilakukan menggunakan metode *Gaussian Naive Bayes*. Langkah terakhir dilakukan pengujian *data test* tersebut dan didapatkan hasil prediksi.

IV. HASIL DAN PEMBAHASAN

A. Normalize Data

Tahap ini adalah tahapan yang penting dalam pra proses *Machine learning*. *Normalize* data berfungsi untuk membuat beberapa *variable* data memiliki rentang nilai yang sama, sehingga tidak ada data yang terlalu besar maupun terlalu kecil.

Pada percobaan ini kami menggunakan metode *Min-Max*, seperti yang terlihat pada Gambar 2. terdapat normalisasi data. Setiap data dikurang dengan data minimal, kemudian dibagi dengan selisih nilai maksimum dan minimum pada fitur tersebut. Sehingga *ouput* dari data ini memiliki *range* antara 0 hingga 1.

```

[[0.70833333 1.      1.      ... 0.      0.      0.33333333]
 [0.16666667 1.      0.66666667 ... 0.      0.      0.66666667]
 [0.25      0.      0.33333333 ... 1.      0.      0.66666667]
 ...
 [0.8125    1.      0.      ... 0.5    0.5    1.      ]
 [0.58333333 1.      0.      ... 0.5    0.25   1.      ]
 [0.58333333 0.      0.33333333 ... 0.5    0.25   0.66666667]]
    
```

Gambar 2. *Normalize data*

B. Pemisahan Data

Data yang digunakan selanjutnya akan dipisah menjadi dua kelompok yaitu data latih dan data tes. Pada Tabel 2. ditampilkan perbandingan persentase dan jumlah dari data latih dan data tes.

TABEL II. PERBANDINGAN DATA LATIH DAN DATA TES

	Data Latih	Data Tes	Total
Persentase	80	20	100
Jumlah	243	61	304

1) Training Data

Training data adalah data yang akan digunakan untuk melatih program yang dibuat nantinya dapat membuat prediksi sehingga mampu mencari korelasi data sendiri atau belajar pola dari data yang diberikan. Berdasarkan pada Gambar 3. Data yang digunakan sebagai data latih sebanyak 80% dari jumlah total data keseluruhan yaitu 304 data sehingga data yang digunakan sebagai data latih berjumlah 242 data.

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
74	0.291667	0.0	0.666667	0.264151	0.198630	0.0	0.5	0.717557	0.0	0.032258	0.5	0.00	0.666667
153	0.770833	0.0	0.666667	0.490566	0.347032	0.0	0.0	0.618321	0.0	0.000000	0.5	0.25	0.666667
64	0.604167	1.0	0.666667	0.433962	0.194064	1.0	0.0	0.717557	0.0	0.000000	1.0	0.00	0.666667
296	0.708333	0.0	0.000000	0.283019	0.162100	0.0	0.5	0.496183	1.0	0.000000	0.5	0.00	0.666667
287	0.583333	1.0	0.333333	0.566038	0.242009	0.0	0.0	0.709924	0.0	0.000000	1.0	0.25	0.666667
...
251	0.291667	1.0	0.000000	0.358491	0.276256	1.0	0.0	0.549618	1.0	0.016129	0.5	1.00	1.000000
192	0.520833	1.0	0.000000	0.245283	0.141553	0.0	0.5	0.320611	0.0	0.225806	0.5	0.25	1.000000
117	0.562500	1.0	1.000000	0.245283	0.152968	0.0	0.0	0.694656	0.0	0.306452	0.5	0.00	1.000000
47	0.375000	1.0	0.666667	0.415094	0.299087	0.0	0.0	0.648855	0.0	0.000000	1.0	0.00	0.666667
172	0.604167	1.0	0.333333	0.245283	0.360731	0.0	0.0	0.679389	0.0	0.290323	0.5	0.00	0.666667

242 rows x 13 columns

Gambar 3. *Training data*

Data latih selanjutnya akan digunakan untuk melatih model dengan menggunakan *Naive Bayes* Metode Gaussian.

```
from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
nb.fit(x_train, y_train)
```

Gambar 4. Kodongan *training data*

Pada Gambar 4 dilakukan penginputan *Naive Bayes* metode Gaussian yang berada di dalam modul *sklearn*. Setelah itu membuat variabel “nb” sebagai pengganti fungsi *GaussianNB()*. Kemudian memasukkan data *training* ke dalam fungsi *training* “fit” untuk melatih model.

2) *Data Tes*

Data tes merupakan data yang digunakan untuk mengetes program yang telah dibuat apakah mampu mencari korelasi data sehingga dapat dilihat keakuratannya. Seperti pada Gambar 5, sisa 20% dari keseluruhan data digunakan sebagai data tes yaitu berjumlah 61 data. Data tes tidak boleh merupakan data yang pernah dilihat oleh model sebelumnya.

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
225	0.854167	1.0	0.000000	0.481132	0.109589	0.0	0.5	0.412214	1.0	0.419355	0.0	0.00	1.000000
152	0.729167	1.0	1.000000	0.716981	0.230594	0.0	0.0	0.641221	0.0	0.096774	0.5	0.00	1.000000
228	0.625000	1.0	1.000000	0.716981	0.369063	0.0	0.0	0.671756	0.0	0.032258	0.5	0.00	1.000000
201	0.645833	1.0	0.000000	0.292453	0.301370	0.0	0.0	0.534351	1.0	0.451613	0.5	0.25	1.000000
52	0.687500	1.0	0.666667	0.339623	0.239726	0.0	0.5	0.572519	0.0	0.290323	0.5	0.75	1.000000
...
146	0.312500	0.0	0.666667	0.226415	0.264840	0.0	0.5	0.595420	0.0	0.048387	0.5	0.25	0.666667
302	0.583333	0.0	0.333333	0.339623	0.251142	0.0	0.0	0.786260	0.0	0.000000	0.5	0.25	0.666667
26	0.625000	1.0	0.666667	0.528302	0.196347	1.0	0.5	0.656489	0.0	0.258065	1.0	0.00	0.666667
108	0.437500	0.0	0.333333	0.245283	0.269406	0.0	0.5	0.694656	0.0	0.177419	1.0	0.00	0.666667
89	0.604167	0.0	0.000000	0.056604	0.278539	0.0	0.0	0.389313	0.0	0.161290	0.5	0.00	0.666667

61 rows x 13 columns

Gambar 5. Data tes

Hasil uji dari model terdapat pada Gambar 6

```
y_test
array([0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1,
       0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0,
       0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0])
```

Gambar 6. Target tes

C. *Data Prediksi*

Data ini adalah data uji yang digunakan untuk melakukan klasifikasi berdasarkan dataset tes. Pada Gambar 7, total dari data prediksi sama dengan data tes yaitu 61 data. Hasil dari data prediksi dapat dilihat sebagai berikut:

```
y_pred = nb.predict(x_test)
y_pred
array([1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1,
       1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1,
       0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0])
```

Gambar 7. Target prediksi

D. *Confusion Matrix*

Confusion Matrix adalah tabel ringkasan perbandingan hasil yang diprediksi dan hasil aktual. Matriks kebingungan dapat digunakan untuk menilai kinerja model klasifikasi.

```

from sklearn.metrics import confusion_matrix
from sklearn.metrics import ConfusionMatrixDisplay

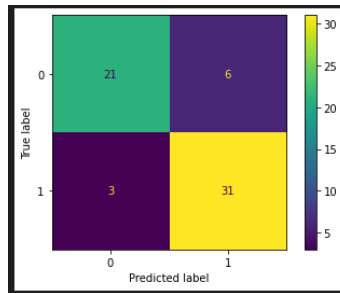
y_pred = nb.predict(x_test)

cm = confusion_matrix(y_test, y_pred)
cm_display = ConfusionMatrixDisplay(cm).plot()

```

Gambar 8. Kodingan *confusion matrix*

Pada Gambar 8 digunakan fungsi *confusion_matrix* dan *ConfusionMatrixDisplay* yang terdapat di dalam modul *sklearn*. "y_pred" merupakan variabel pengganti fungsi *nb.predict()* yang digunakan untuk menguji model menggunakan data yang akan diuji. Selanjutnya dilakukan pengecekan hasil data uji dengan "hasil yang sebenarnya" menggunakan fungsi *confusion_matrix*. Variabel "cm_display" digunakan untuk menggantikan fungsi *ConfusionMatrixDisplay().plot()* yang berfungsi untuk menampilkan hasil dari *confusion_matrix*.

Gambar 9. *True label* dan *predicted label*

Dari Gambar 9, matriks kebingungan menunjukkan bahwa 52 data yang terprediksi dinyatakan benar sesuai dengan "hasil yang sebenarnya". Di antaranya sebanyak 21 data yang terprediksi benar tidak mengidap penyakit jantung dan data yang terprediksi benar mengidap sakit jantung berjumlah 31 data. Kemudian terdapat 6 data terprediksi mengidap sakit jantung dan 3 data terprediksi tidak mengidap penyakit jantung adalah salah. Sehingga total data yang terprediksi salah berjumlah 9 data.

E. Tingkat Akurasi

Dari 61 data yang diprediksi, data yang benar berjumlah 52 sedangkan data yang salah berjumlah 9 buah data. Sehingga nilai akurasi dari model ini menggunakan persamaan (3) adalah sebagai berikut:

$$\begin{aligned}
 \text{Akurasi} &= \frac{\Sigma \text{Prediksi Benar}}{\Sigma \text{Prediksi}} \times 100\% & (3) \\
 &= \frac{52}{61} \times 100\% \\
 &= 85.2459\% \\
 &= 85.25\%
 \end{aligned}$$

Nilai ini sesuai dengan nilai yang terdapat pada program di bawah ini.

```

from sklearn.metrics import accuracy_score
acc = accuracy_score(y_test, y_pred)*100
accuracies['Naive Bayes'] = acc
print("Akurasi: {:.2f}%".format(acc))
✓ 0.3s
Akurasi: 85.25%

```

Gambar 10. Hasil akurasi

Pada gambar 10. dilakukan penginputan perhitungan untuk menentukan nilai dari akurasi (*accuracy_score*) yang terdapat didalam modul *sklearn*. Setelah itu dibuat variabel "acc" sebagai pengganti dari fungsi *accuracy_score* yang digunakan untuk menghitung akurasi dari data yang telah diuji. Pada perhitungan akurasi ini, hasilnya berupa nilai 1 angka dibelakang koma sehingga dikalikan dengan 100 untuk menampilkan bilangan puluhan. Fungsi *print()* digunakan untuk menampilkan nilai akurasi dengan mengambil nilainya sampai ke 2 angka di belakang koma.

V. KESIMPULAN DAN SARAN

Model yang dibuat menggunakan metode *Naive Bayes* mampu mendiagnosa apakah seseorang mengidap penyakit jantung. Dari 61 data yang diprediksi, data yang terprediksi benar berjumlah 52 data. Di antaranya sebanyak 21 data yang terprediksi benar tidak mengidap penyakit jantung dan data yang terprediksi benar mengidap sakit jantung berjumlah 31 data. Kemudian terdapat 6 data terprediksi mengidap sakit jantung dan 3 data terprediksi tidak mengidap penyakit jantung adalah salah. Sehingga total data yang terprediksi salah berjumlah 9 data. Sehingga tingkat keakuratannya sebesar 85.25%. Nilai tersebut terbilang tinggi karena dataset yang digunakan berjumlah 304 dan data yang digunakan sebagai data latih berjumlah 242 data.

Berdasarkan penelitian yang telah dilakukan, penulis menyarankan untuk sistem prediksi penyakit dapat dicoba dengan menggunakan metode klasifikasi lain agar dapat dilakukan perbandingan metode apa yang lebih efektif untuk memprediksi penyakit jantung.

DAFTAR PUSTAKA

- [1] A. Gilbert, H. Nabila, W. Andre dan P. Jeremy, "Perbandingan Pola Sinyal Penyakit Myocardial Infarction dengan Jantung Normal Menggunakan Metode Wavelet Symlet," *ULTIMA Computing*, vol. XIII, no. 1, pp. 49-56, Jun. 2020.
- [2] G. Subbalakshmi, K. Ramesh dan C. Rao, "Decision Support in Heart Disease Prediction System using Naive Bayes," *Indian Journal of Computer Science and Engineering (IJCSE)*, vol. I, no. 2, p. 170-176, 2011.
- [3] S. Kumar, D. Shah dan S. Patel, "Heart Disease Prediction using Machine Learning Techniques," *SN Computer Science*, vol. I, no. 1, p. 345, Oct. 2020.
- [4] C. Nagesh Singh, "KD Nuggets," Juni 2020. [Online]. Available: <https://www.kdnuggets.com/2020/06/naive-bayes-algorithm-everything.html>. [Accessed 9 November 2021].
- [5] Anonim, "Java T Point," Noida, [Online]. Available: <https://www.javatpoint.com/machine-learning-naive-bayes-classifier>. [Accessed 9 November 2021].
- [6] K. Vembandasamy, R. Sasipriya and E. Deepa, "Heart Diseases Detection Using Naive Bayes Algorithm," *International Journal of Innovative Science, Enginnering & Technology (IJSET)*, vol. II, no. 9, pp. 441-444, Sep. 2015.
- [7] A. Janosi, W. Steinbrunn, M. Pfisterer and R. Detrano, "UCI Machine Learning Repository," University of California, Irvine, [Online]. Available: <https://archive.ics.uci.edu/ml/support/Heart%20Disease>. [Accessed 7 November 2021].