

# Analisis Kinerja Mikrokomputer Raspberry Pi Pada *Smart Greenhouse* Berbasis *Internet Of Things (IoT)* Menggunakan Algoritma *Naive Bayes*

## *Performance Analysis Of Raspberry Pi Microcomputer In Internet Of Things (IoT)-Based Smart Greenhouse Using Naive Bayes Algorithm*

Philipus Asten<sup>1)</sup>, Darsono Nababan <sup>\*,2)</sup> Risald <sup>\*,3)</sup>

Program Studi Teknologi Informasi, Fakultas Sains dan Teknik, Universitas Timor  
Jl. El Tari Km.05, Kefamenanu- NTT, Indonesia 50275

Riwayat : Copyright ©2023, JITU, Submitted : 20 September 2023; Revised: 22 September 2023; Accepted: 22 September 2023; Published: 30 September 2023  
DOI : 10.32938/jitu.v3i2.5231

**Abstrack** - *Smart Greenhouse* is a system that utilizes *Internet of Things (IoT)* technology to automatically control and monitor plant growth environment. In analyzing the performance of the *Raspberry Pi* microcomputer in the *IoT*-based *Smart Greenhouse*, the *Naive Bayes* algorithm is applied to analyze data and make decisions based on plant environmental conditions. The analysis results of the *Raspberry Pi* microcomputer in the *Smart Greenhouse* using the *Naive Bayes* algorithm can be used to predict environmental variables such as temperature and humidity, based on data obtained from a total dataset of 500 rows, using 4 variables. The evaluation results for air temperature are *MSE* (0.25) and *RMSE* (0.50), for air humidity *MSE* (0.07) and *RMSE* (0.27), for soil humidity *MSE* (0.00) and *RMSE* (0.00), and *UV* with *MSE* (0.94) and *RMSE* (0.97).

**Keywords:** *Raspberry Pi*, *smart greenhouse*, *Internet of Things (IoT)*, *Naive Bayes* algorithm, performance, analysis."

**Abstrak** - *Smart Greenhouse* adalah sistem yang menggunakan teknologi *Internet of Things (IoT)* untuk mengontrol dan memantau lingkungan pertumbuhan tanaman secara otomatis. Dalam melakukan analisis kinerja mikrokomputer *Raspberry Pi* dalam *Smart Greenhouse* berbasis *IoT*, dengan menerapkan algoritma *Naive Bayes* untuk menganalisis data dan membuat keputusan berdasarkan kondisi lingkungan tanaman. Hasil analisis terhadap mikrokomputer *Raspberry Pi* pada *Smart Greenhouse* menggunakan algoritma *Naive Bayes* dapat digunakan untuk memprediksi variabel lingkungan, seperti suhu dan kelembapan, sesuai data yang diperoleh dengan total data set adalah 500 baris menggunakan 4 variabel dengan mempunyai nilai evaluasi terhadap suhu udara dengan *MSE* (0.25) dan *RMSE* (0.50), kelembapan udara dengan *MSE* (0.07) dan *RMSE* (0.27), kelembapan tanah

dengan *MSE* (0.00) dan *RMSE* (0.00) dan *UV* dengan nilai *MSE* (0.94) dan *RMSE* (0.97)

**Kata kunci:** *Raspberry Pi*, *smart greenhouse*, *Internet of Things (IoT)*, algoritma *naive bayes*, kinerja, analisis.

### I. PENDAHULUAN

Berkembangnya ilmu pengetahuan dan teknologi saat ini memberikan pengaruh besar dalam kehidupan manusia di masa sekarang hampir semua aspek kehidupan tidak lepas dari pengaruh perkembangan teknologi ini termasuk aspek pertanian. Di era modern ini kita tidak asing lagi dengan dengan *smart greenhouse*, dengan penerapan *smart greenhouse* ini diharapkan petani akan terbantu dan dipermudah untuk mencapai target yang diinginkan dengan alih fungsi lahan pertanian semakin meningkat oleh manusia yang digunakan sebagai tempat tinggal, usaha dan fasilitas-fasilitas umum sehingga sebagian besar lahan pertanian semakin terbatas agar pengalihan lahan pertanian ini menjadikan keseimbangan ekosistem menjadi terganggu [1].

Selain merancang bangun *smart greenhouse* berbasis *Internet of Things (IoT)* dengan menggunakan *Raspberry Pi* harus dapat di analisis menggunakan algoritma *Naive Bayes* untuk mengetahui kinerja sistem *smart greenhouse* apakah kinerjanya maksimal atau tidak [2]. Selain itu untuk prediksi keadaan di masa mendatang melalui pengujian di masa lalu. Penggunaan metode *naive bayes* ini tentunya membantu mahasiswa untuk melihat performa alat di tahap penerapan berikutnya, sehingga mahasiswa lebih mudah untuk memutuskan apakah tetap menggunakan alat-alat tersebut atau melakukan perbaikan maupun mengganti komponen-komponen tersebut ke depannya [3].

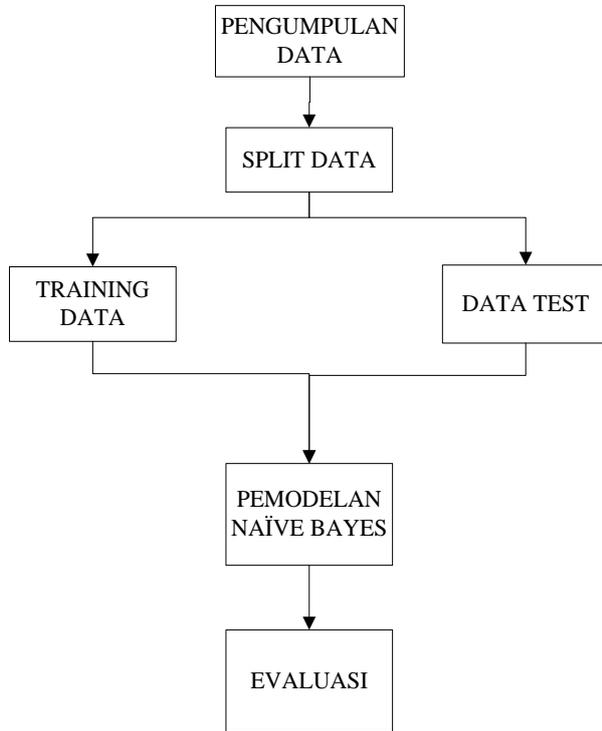
Dengan algoritma *Naive Bayes* pada *smart greenhouse* berbasis *Internet of Things (IoT)* menggunakan sebuah cara untuk menemukan properti yang sama pada himpunan objek di dalam sebuah data pada *smart greenhouse* dan mengklasifikasikannya ke dalam kelas-kelas yang berbeda menurut model klasifikasi yang di tetapkan bertujuan untuk menemukan akurasi dari kinerja *smart greenhouse*.

\*) Penulis korespondensi (Nama Penulis)  
Email: corr\_author@institut.ac.id

## II. METODE PENELITIAN

### 1. Metode penelitian

Dalam menganalisis kinerja mikrokomputer Raspberry Pi pada *Smart Greenhouse* berbasis *Internet of Things*. Teknik yang digunakan dalam penelitian ini menggunakan algoritma *Naive Bayes* [4].



**Gambar 1.** Pemodelan Naive Bayes

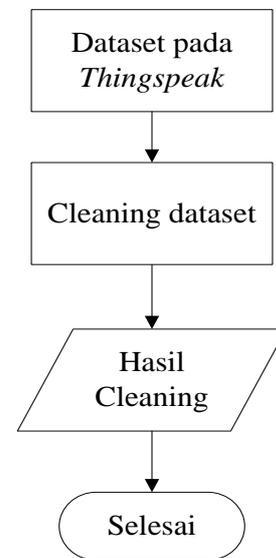
- Pengumpulan Data yang telah dilakukan melalui proses *pre-processing* yaitu mulai dari Inialisasi, baca data sensor *Soil Moisture*, DHT22 dan GUVAS12SD, Raspberry PI memproses data, kemudian ESP8266 mengirim data ke *server (thing speak)*, kemudian melakukan *cleaning* data sensor *Soil (Soil Moisture, DHT22 dan GUVAS12SD)* untuk melakukan analisis menggunakan algoritma *naive bayes*.
- Split Data* atau pemisahan data adalah metode membagi data menjadi dua bagian atau lebih yang membentuk sub himpunan data. Umumnya, data *splitting* memisahkan dua bagian, bagian pertama digunakan untuk mengevaluasi atau uji data dan data lainnya digunakan untuk melatih model
- Training Data* atau data latih digunakan untuk melatih dan mengembangkan model. Kumpulan data *training* biasanya digunakan untuk mengestimasi parameter yang berbeda atau untuk membandingkan kinerja model yang berbeda.
- Data *Test* atau data uji digunakan setelah proses *training* selesai. Data *training* dan testing dibandingkan untuk memeriksa apakah model akhir yang digunakan bekerja dengan benar.
- Algoritma *Naive Bayes* merupakan sebuah metode

klasifikasi yang berakar pada pengklasifikasian dengan menggunakan metode probabilitas dan statistic [3]. *Naive Bayes* ini adalah asumsi yang sangat kuat akan independensi dari masing-masing kondisi atau kejadian. Untuk setiap kelas keputusan, menghitung probabilitas dengan syarat bahwa kelas keputusan adalah benar, mengingat vektor informasi obyek.

- Evaluasi menjadi hal yang sangat penting untuk mengetahui tingkat akurasi, dari data yang analisis menggunakan *naive bayes*.

### 2. Data Penelitian

*Dataset* diperoleh dari data perhitungan sensor selama satu minggu yang kemudian di simpan pada *server thingspeak*, yang dihasilkan melalui alat dengan waktu pukul 08:00-16:00 WITA selama 1 minggu serta satuan untuk suhu adalah derajat *celcius*, kelembapan adalah *relative humidity*, dan intensitas sinar *uv* adalah nano meter. Berikut adalah alur untuk perolehan data-datanya.



**Gambar 2.** Tahapan pemrosesan data

## III. HASIL DAN PEMBAHASAN

### 1. Proses Analisis Data Naive Bayes

Alur pengerjaan analisis dari *dataset time series* yang di hitung sensor dan sudah melalui tahapan *cleaning*, kemudian dilakukan analisis menggunakan bahasa pemrograman python dengan jupyter notebook sebagai teks editornya. Berikut *workflow* dari analisis *time series*.



Gambar 3. Tahapan Penelitian

## 2. Analisis Menggunakan Naïve Bayes

### A. Preprocessing Data

Dalam memanggil data dalam Jupyter Notebook hal pertama yang perlu dilakukan adalah *import library*, tujuannya agar dapat digunakan berulang kali dalam program yang berbeda. Perintah yang dapat digunakan sebagai berikut:

```

Import Library
numpy as np
import pandas as pd
import datetime
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.naive_bayes import GaussianNB
from sklearn.cluster import KMeans
from sklearn.metrics import classification_report, confusion_matrix
  
```

### B. Menampilkan 5 data awal

Tabel 1. Data 5 baris pertama

	Suhu Udara	Kelembapan Udara	Uv	Kelembapan Tanah
Date				
2022-08-30	34	99	91	255
2022-08-30	34	99	95	255
2022-08-30	34	99	95	255
2022-08-30	34	99	81	255
2022-08-30	34	99	71	255

### C. Menampilkan 5 data akhir

Tabel 2. Baris Data 5 terakhir

	Suhu Udara	Kelembapan Udara	Uv	Kelembapan Tanah
Date				
2022-08-30	34	99	91	255
2022-08-30	34	99	95	255
2022-08-30	34	99	95	255
2022-08-30	34	99	81	255
2022-08-30	34	99	71	255

### D. Mengubah Data Menjadi Bentuk *Time Series*

Pada tahap ini data akan di ubah dalam bentuk *time series* dengan tujuan untuk membantu mempelajari data pada waktu sebelumnya agar dapat mempertimbangkan data di masa yang akan datang atau memprediksi data kedepannya. Berikut kita gunakan perintah bentuk *time series*:

```

n_in=1
n_out=1
cols = list()
# input sequence (t-n, ... t-1)
for i in range(n_in, 0, -1):
    cols.append(data.shift(i))
# forecast sequence (t, t+1, ... t+n)
for i in range(0, n_out):
    cols.append(data['LABEL'].shift(-i))
    cols.append(df['LABEL SUHU UDARA'].shift(-i))
    cols.append(df['LABEL KELEMBAPAN UDARA'].shift(-i))
    cols.append(df['LABEL KELEMBAPAN TANAH'].shift(-i))
    cols.append(df['LABEL UV'].shift(-i))
  
```

### E. Split Data

*Split* data yang digunakan saat ini untuk memisahkan data atau membagi data dari keseluruhan menjadi

data yang siap di analisis menggunakan perintah sebagai berikut:

```
gg_500 = agg.sample(500)
agg_500
```

#### F. Normalisasi *Dataset*

Pada tahapan ini data akan di normalisasi dengan tujuan agar dapat mengubah kolom numerik dalam himpunan data untuk menggunakan skala umum dengan perintah sebagai berikut:

```
# normalize the dataset
X_scaled =
scaler.fit_transform(X_1,X_2,X_3,X_4)
```

#### G. Pemodelan Algoritma *Naïve Bayes*

Pada tahapan ini data yang akan di analisis akan di *input* secara otomatis ke dalam model *naïve bayes* untuk mengetahui seberapa besar nilai akurasi, serta nilai prediksi dari data yang di analisis dengan menggunakan perintah sebagai berikut:

```
pred_y_1, y_2, y_3, y_4 =
model_suhu.predict(X_scaled)
```

### 3. Evaluasi

#### A. Suhu Udara

Salah satu cara untuk mengimplementasikan tingkat akurasi hasil testing adalah melakukan evaluasi terhadap suhu udara menggunakan label suhu udara. Evaluasi ini akan menggunakan *Mean Squared Error (MSE)* dan *Root Mean Squared Error (RMSE)* di mana fungsi dari *MSE* yaitu menghitung rata-rata kesalahan kuadrat antara nilai aktual dan nilai peramalan. Sedangkan untuk *RMSE* berfungsi untuk mengevaluasi teknik peramalan yang digunakan untuk mengukur tingkat akurasi hasil perkiraan suatu model.

Berikut adalah perintah yang digunakan dalam melakukan evaluasi model *naïve bayes*:

```
mse =
mean_squared_error(np.array(
[int(i) for i in
y_1.values]),np.array([int(i)
] for i in pred_y_1)),)
rmse =
mean_squared_error(np.array(
[int(i) for i in
y_1.values]),np.array([int(i)
] for i in pred_y_1)),
squared=False)
print('MSE Train: %.2f'%mse)
print('RMSE Train: %.2f'%rmse)
```

```
MSE Train: 0.25
RMSE Train: 0.50
```

Dari hasil evaluasi tersebut didapatkan nilai *Mean Squared Error (MSE)* dan *Root Mean Squared Error (RMSE)* dari data suhu udara, dengan nilai *MSE* = 0.25 dan nilai *RMSE* = 0.50. Dengan demikian performa data suhu

udara yang rekam oleh sensor dapat dikatakan cukup baik dan masih bisa digunakan dalam tahap berikutnya.

#### B. Kelembapan Udara

Salah satu cara untuk mengimplementasikan tingkat akurasi hasil testing adalah melakukan evaluasi terhadap kelembapan udara menggunakan label kelembapan udara. Evaluasi ini akan menggunakan *Mean Squared Error (MSE)* dan *Root Mean Squared Error (RMSE)* di mana fungsi dari *MSE* yaitu menghitung rata-rata kesalahan kuadrat antara nilai aktual dan nilai peramalan. Sedangkan untuk *RMSE* berfungsi untuk mengevaluasi teknik peramalan yang digunakan untuk mengukur tingkat akurasi hasil perkiraan suatu model.

Berikut adalah perintah yang digunakan dalam melakukan evaluasi model *naïve bayes*:

```
print('MSE Train: %.2f'%mse)
print('RMSE Train: %.2f'%rmse)
```

```
MSE Train: 0.07
RMSE Train: 0.27
```

Dari hasil evaluasi tersebut didapatkan nilai *Mean Squared Error (MSE)* dan *Root Mean Squared Error (RMSE)* terhadap kelembapan udara, dengan nilai dari *MSE* = 0.07 dan nilai *RMSE* = 0.27. Dengan demikian performa data suhu udara yang rekam oleh sensor dapat dikatakan cukup baik dan masih bisa digunakan dalam tahap berikutnya.

#### C. Kelembapan Tanah

Salah satu cara untuk mengimplementasikan tingkat akurasi hasil testing adalah melakukan evaluasi terhadap kelembapan tanah menggunakan label kelembapan tanah. Evaluasi ini akan menggunakan *Mean Squared Error (MSE)* dan *Root Mean Squared Error (RMSE)* di mana fungsi dari *MSE* yaitu menghitung rata-rata kesalahan kuadrat antara nilai aktual dan nilai peramalan. Sedangkan untuk *RMSE* berfungsi untuk mengevaluasi teknik peramalan yang digunakan untuk mengukur tingkat akurasi hasil perkiraan suatu model.

Berikut adalah perintah yang digunakan dalam melakukan evaluasi model *naïve bayes*:

```
print('MSE Train: %.2f'%mse)
print('RMSE Train: %.2f'%rmse)
```

```
MSE Train: 0.00
RMSE Train: 0.00
```

Dari hasil evaluasi tersebut didapatkan nilai *Mean Squared Error (MSE)* dan *Root Mean Squared Error (RMSE)* terhadap kelembapan tanah, dengan nilai dari *MSE* = 0.00 dan nilai *RMSE* = 0.00. Dengan demikian performa data suhu udara yang rekam oleh sensor dapat dikatakan sangat baik dan masih bisa digunakan dalam tahap berikutnya.

#### D. UV

Salah satu cara untuk mengimplementasikan tingkat akurasi hasil testing adalah melakukan evaluasi terhadap *UV* menggunakan label *UV*. Evaluasi ini akan menggunakan *Mean Squared Error (MSE)* dan *Root Mean Squared Error (RMSE)* di mana fungsi dari *MSE* yaitu menghitung rata-rata kesalahan kuadrat antara nilai aktual dan nilai peramalan. Sedangkan untuk *RMSE* berfungsi untuk mengevaluasi teknik peramalan yang digunakan untuk mengukur tingkat akurasi hasil perkiraan suatu model. Berikut adalah perintah yang digunakan dalam melakukan evaluasi model *naive bayes*:

```
print('MSE Train: %.2f'%mse)
print('RMSE Train: %.2f'%rmse)

MSE Train: 0.94
RMSE Train: 0.97
```

Dari hasil evaluasi tersebut didapatkan nilai *Mean Squared Error (MSE)* dan *Root Mean Squared Error (RMSE)* terhadap *UV*, dengan nilai dari *MSE* = 0.94 dan nilai *RMSE* = 0.97. Dengan demikian performa data suhu udara yang rekam oleh sensor dapat dikatakan cukup baik dan masih bisa digunakan dalam tahap berikutnya.

#### 4. Perhitungan Manual

Salah satu algoritma yang sering digunakan untuk *data mining* adalah *Naïve Bayes*, dikarenakan *Naïve Bayes* memiliki performa yang kompetitif untuk proses memprediksi walaupun tidak ada keterkaitan antara atribut. Metode *Naïve Bayes* merupakan pengelompokan data statistik atau perhitungan manual sebagai hasil perbandingan dengan perhitungan menggunakan python. Berikut merupakan tahapan perhitungan:

##### A. Prediksi Data

Dalam melakukan Peramalan bertujuan untuk mengetahui keadaan di masa depan dengan cara menghitung dari data-data yang diambil sebagai acuan sehingga bisa meramalkan ke beberapa variabel [5] yang dihitung sensor menggunakan rumus sebagai berikut:

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

hasil dari peramalan dapat menghasilkan tabel berikut

**Tabel 3.** Hasil perhitungan Naive Bayes

No	DATE	PRED SUHU UDARA	PRED KELEMBAPAN UDARA	PRED UV	PRED KELEMBAPAN TANAH
1	2022-08-30	35	99	92	255
2	2022-08-30	35	99	96	255
3	2022-08-30	35	99	96	255

4	2022-08-30	35	99	82	255
5	2022-08-30	35	99	72	255
...	...	...	...	...	...
...	...	...	...	...	...
...	...	...	...	...	...
496	2022-09-04	39	76	57	255
497	2022-09-04	38	78	52	255
498	2022-09-04	39	58	54	255
499	2022-09-04	38	60	56	255
500	2022-09-04	38	65	50	255

##### B. Menentukan Nilai Error (MSE dan RMSE)

Dari data hasil prediksi diatas maka akan dilakukan evaluasi dengan menentukan nilai *Mean Squared Error (MSE)* untuk menentukan akurasi peramalan dari sebuah kesalahan (*error*) [6].

$$C. \quad MSE = \frac{1}{n} \sum_{t=1}^n \{Y_t - Y'_t\}^2 \quad (2)$$

Dengan hasil *MSE* dapat diketahui nilai *Root Mean Squared Error (RMSE)* merupakan metode pengukuran dengan mengukur perbedaan nilai dari prediksi sebuah model sebagai estimasi atas nilai yang diobservasi [7] yang di menggunakan rumus:

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n \{Y_t - Y'_t\}^2} \quad (3)$$

dari hasil perhitungan diatas dapat diketahui dengan nilai evaluasi sebagai berikut:

- Suhu Udara  
MSE : 0.25  
RMSE : 0.50
- Kelembapan Udara  
MSE : 0.07  
RMSE : 0.27
- Kelembapan Tanah  
MSE : 0.00  
RMSE : 0.00
- UV*  
MSE : 0.94  
RMSE : 0.97

#### IV. KESIMPULAN

Untuk menganalisis kinerja mikrokomputer dengan algoritma *Naïve Bayes* yang telah dilakukan menghasilkan kesimpulan bahwa algoritma *Naïve Bayes* dapat menentukan hasil prediksi dari suatu review ke dalam dua kategori yaitu positif yang diimplementasikan dalam kata *satisfied* atau puas dan juga *un satisfied* atau tidak puas. Berdasarkan hasil pengujian yang dilakukan, hasil prediksi data suhu udara, kelembaban udara, kelembapan tanah dan *UV* dengan algoritma *naïve bayes* memberikan *MSE* pada suhu udara sebesar = 0.25 dan *RMSE* sebesar = 0.50, kelembaban udara *MSE* sebesar = 0.07 dan *RMSE* sebesar = 0.27, kelembaban tanah *MSE* sebesar = 0.00 dan *RMSE* sebesar = 0.00 kemudian kolom *UV* dengan *MSE* sebesar = 0.94 dan *RMSE* sebesar = 0.97 maka hasil prediksi menggunakan metode *Naive Bayes* menunjukkan kinerja yang baik jika dilihat dari akurasi data yang di dapatkan.

#### DAFTAR PUSTAKA

- [1] S. R. Hartanto, R. Ulansari, and A. Syakur, "674-2239-1-Pb," vol. 5, no. 2, 2019.
- [2] K. A. Sandy, A. Aribowo, A. S. Putra, and A. R. Mitra, "Sistem Penyiraman Otomatis," vol. 11, no. 1, pp. 7–15, 2021.
- [3] V. A. Permadi, "Analisis Sentimen Menggunakan Algoritma Naive Bayes Terhadap Review Restoran di Singapura," *J. Buana Inform.*, vol. 11, no. 2, pp. 141–151, 2020, doi: 10.24002/jbi.v11i2.3769.
- [4] T. A. Pratiwi, M. Irsyad, R. Kurniawan, S. Agustian, and B. S. Negara, "Klasifikasi Kebakaran Hutan Dan Lahan Menggunakan Algoritma Naïve Bayes Di Kabupaten Pelalawan," *CESS (Journal Comput. Eng. Syst. Sci.)*, vol. 6, no. 1, p. 139, 2021, doi: 10.24114/cess.v6i1.22555.
- [5] Y. E. Fadrial, "Algoritma Naive Bayes Untuk Mencari Perkiraan Waktu Studi Mahasiswa," *INTECOMS J. Inf. Technol. Comput. Sci.*, vol. 4, no. 1, pp. 20–29, 2021, doi: 10.31539/intecom.v4i1.2219.
- [6] A. Prasetya Wibawa, W. Lestar, A. Bella Putra Utama, I. Tri Saputra, and Z. Nabila Izdihar, "Multilayer Perceptron untuk Prediksi Sessions pada Sebuah Website Journal Elektronik," *Indones. J. Data Sci.*, vol. 1, no. 3, pp. 57–67, 2020, doi: 10.33096/ijodas.v1i3.15.
- [7] I. Rasila, U. Ristian, J. Rekeyasa Sistem Komputer, and F. H. MIPA Universitas Tanjungpura Jl Hadari Nawawi, "Implementasi Metode Naive Bayes Classifier Pada Sistem Pengklasifikasi Berita Otomatis Berbasis Website (Studi Kasus: Berita Lokal Dari Mediamassa Online Kalimantan Barat)," *Coding J. Komput. dan Apl.*, vol. 07, no. 2, pp. 49–60, 2019.