INOTEK, Vol.8 Agustus 2024
ISSN: 2580-3336 (Print) / 2549-7952 (Online)
Url: https://proceeding.unpkediri.ac.id/index.php/inotek/

Analisis Perbandingan Algoritma Naïve Bayes dengan K-Nearest Neighbor (KNN) Pada Dataset Mobile Price Classification

Diterima:
10 Juni 2024
Revisi:
10 Juli 2024
Terbit:
1 Agustus 2024

[1*] Ovelina Devi Kurnia, [2]Elisa Triammah A'Fena, [3]Dea Yuliana Ayu Ningrum, [4] Erna Daniati, [5] Aidina Ristyawan
[1-5]Universitas Nusantara PGRI Kediri
ovelina.dk@gmail.com, : ernadaniati@unpkediri.ac.id

Abstrak— Penelitian ini berfokus pada penerapan data mining untuk Analisis perbandingan algoritma *Naïve Bayes* dengan *K-Nearest Neighbor* (KNN) dengan dataset klasifikasi harga *smartphone* menggunakan *Jupyter*. Dalam penelitian ini, kita membandingkan kinerja dua algoritma klasifikasi, *Naïve Bayes* dan KNN, dengan tujuan untuk memprediksi kisaran harga yang menunjukkan seberapa tinggi harga tersebut berdasarkan fitur-fitur yang tersedia. Pada penelitian acuan jurnal tingkat akurasinya antara *naive bayes* maupun kNN termasuk rendah. Dan pada penelitian kali ini menunjukkan hasil bahwa KNN memiliki akurasi yang lebih tinggi dibandingkan dengan *Naïve Bayes* dalam memprediksi harga smartphone.
Kata Kunci— Data Mining; *Naïve bayes*; Klasifikasi; KNN

Abstract— *This study focuses on the application of data mining for Comparative analysis of the Naïve Bayes algorithm with K-Nearest Neighbor (KNN) with a smartphone price classification dataset using Jupiter. In this study, we compare the performance of two classification algorithms, Naïve Bayes and KNN, with the aim of predicting the price range that indicates how high the price is based on the available features. In the journal reference study, the accuracy level between naive bayes and kNN is low. And this study shows that KNN has a higher accuracy compared to Naïve Bayes in predicting smartphone prices.*
Keywords— *Data Mining; Naïve bayes; Classification; Quality of Service*

Penulis Korespondensi:

Erna Daniati,
Sistem Informasi,
Universitas Nusantara PGRI Kediri,
Email: ernadaniati@unpkediri.ac.id
ID Orcid: [https://orcid.org/0009-0008-9471-4421]
Handphone: +62 813-3524-2202

I. PENDAHULUAN

Teknologi saat ini berkembang dengan sangat cepat, terutama dalam hal pengolahan data. Kemajuan teknologi dapat mengakibatkan tersedianya data dan informasi yang dapat digunakan untuk mengambil keputusan. Menggunakan teknik data mining adalah salah satu metode yang dapat digunakan jika Anda memiliki banyak informasi. Untuk memberikan informasi yang lebih tepat dan bermanfaat, data dapat diolah dan diperiksa. Oleh karena itu, teknik data mining menjadi sangat penting di era digital yang kaya akan data.

Menganalisis kumpulan data untuk menemukan hubungan yang tidak terduga dan menyajikan informasi dengan cara yang berbeda dari metode sebelumnya, namun tetap dapat dimengerti dan bermanfaat bagi pemilik data, dikenal sebagai penggalian data. Untuk memecahkan masalah penggalian informasi dari database yang sangat besar, disiplin ilmu data mining menggabungkan metode-metode dari pembelajaran mesin, pengenalan pola, statistik, database, dan visualisasi[1].

Menemukan model atau fungsi yang mendefinisikan dan memisahkan ide dan kelas data adalah proses klasifikasi data. Untuk mengidentifikasi algoritme dengan akurasi terbaik, para peneliti akan membandingkan sejumlah algoritme[2]. Melalui penggunaan teknologi pengenalan pola, metode statistik dan matematika, dan volume data yang sangat besar yang disimpan di repositori, penggalian data adalah proses menemukan korelasi, pola, dan tren baru yang relevan[3].

Naive Bayes merupakan metode klasifikasi sederhana yang menghitung semua probabilitas berdasarkan teorema Bayes yang dikombinasikan dengan kombinasi nilai frekuensi *database* [4]. Menurut Olson Delen(2008) melalui penggunaan teknologi pengenalan pola, metode statistik dan matematika, dan volume data yang sangat besar yang disimpan di repositori, penggalian data adalah proses menemukan korelasi, pola, dan tren baru yang relevan[5]. Kelebihan dari *naive bayes* yaitu mudah diimplementasikan, lebih cepat dalam perhitungan dan memerlukan jumlah data sedikit yang dibutuhkan untuk klasifikasi[6].

Akronim *Jupyter* mewakili tiga bahasa pemrograman: R, Python (Py), dan Julia (Ju). Ilmuwan data paling sering menggunakan *Jupyter Notebook*, sebuah aplikasi *online* gratis, yang paling sering digunakan. Program ini memungkinkan Anda untuk membuat dan berbagi dokumen dengan teks, kode, grafik, dan hasil perhitungan[7]. *Jupyter notebook* merupakan lingkungan yang cukup interaktif dalam menjalankan baris perintah berupa kode pada *browser* dan termasuk *tools* yang berguna dalam membuat sebuah pekerjaan yang berhubungan dengan analisis ilmuwan data[8].

Smartphone telah menjadi bagian integral dari kehidupan manusia. Berbagai merek dan model *smartphone* tersedia di pasaran, dengan harga yang berbeda-beda. Oleh karena itu, klasifikasi harga *smartphone* menjadi salah satu masalah yang penting dalam bisnis teknologi. Dalam penelitian ini, kita akan membandingkan kinerja dua algoritma klasifikasi, *Naïve Bayes* dan *K-Nearest Neighbor* (KNN), untuk memprediksi harga *smartphone* berdasarkan fitur-fitur yang tersedia.

II. METODE PENELITIAN

2.1 Algoritma *naïve bayes*

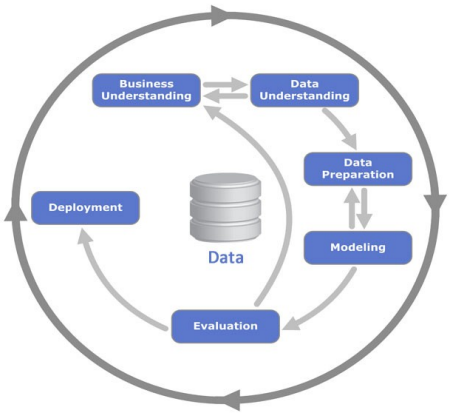
Algoritma yang digunakan pada penelitian jurnal ini adalah algoritma *Naïve Bayes Classifier.* Metode pengklasifikasian dengan menggunakan metode probabilitas dan statistik, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai Teorema Bayes[9]. Proses klasifikasi dengan *Naïve Bayes* dilakukan menggunakan data latih yang sebelumnya sudah dibagi menggunakan *k-fold cross validation*[10].

2.2 *K-Nearest Neighbor* (KNN)

Karena metode KNN menggunakan pengawasan, metode ini membutuhkan data pelatihan untuk mengidentifikasi objek berdasarkan jarak terdekat. Metode dasar KNN adalah menggunakan nilai k untuk menentukan jarak terdekat[11].

2.3 CRISP-DM

CRISP-DM (*Cross Industry Standard Process for Data Mining*) adalah suatu standarisasi pemrosesan data mining yang dirancang agar data yang ada melewati setiap langkah terstruktur dan terdefinisi dengan baik dan efisien[12]. Metode ini terdiri dari enam fase berulang seperti terlihat pada Gambar 2. Fase pertama dimulai dari *Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation, dan Deployment*[13]. Tujuan dari metode CRISP-DM ini adalah untuk melakukan proses analisis strategi yang digunakan untuk memecahkan masalah penelitian ataupun permasalahan dari sebuah bisnis atau perusahaan[14]. Metodologi penelitian CRISPM-DM (*Cross-Industry Standard Process For* Data Mining) yang dilakukan terdiri dari 6 fase yaitu *business understanding, data understanding, data preparation, modelling, evaluation, dan deployment*[15]. Pada Gambar 01 merupakan tahapan dalam proses CRISP-DM.



Gambar 01. Alur proses CRISP-DM

III. HASIL DAN PEMBAHASAN

3.1 Pembahasan penelitian ini dimulai dari tahapan pertama dalam proses CRISP-DM yaitu data *understanding*.

Out[165]:

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	px_height	px_width	ram	sc_h	sc_w	talk_time
0	842	0	2.2	0	1	0	7	0.6	188	2	...	20	756	2549	9	7	19
1	1021	1	0.5	1	0	1	53	0.7	136	3	...	905	1988	2631	17	3	7
2	563	1	0.5	1	2	1	41	0.9	145	5	...	1263	1716	2603	11	2	9
3	615	1	2.5	0	0	0	10	0.8	131	6	...	1216	1786	2769	16	8	11
4	1821	1	1.2	0	13	1	44	0.6	141	2	...	1208	1212	1411	8	2	15
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1995	794	1	0.5	1	0	1	2	0.8	106	6	...	1222	1890	668	13	4	19
1996	1965	1	2.6	1	0	0	39	0.2	187	4	...	915	1965	2032	11	10	16
1997	1911	0	0.9	1	1	1	36	0.7	108	8	...	868	1632	3057	9	1	5
1998	1512	0	0.9	0	4	1	46	0.1	145	5	...	336	670	869	18	10	19
1999	510	1	2.0	1	5	1	45	0.9	168	6	...	483	754	3919	19	4	2

2000 rows × 21 columns

Gambar 02. Tampilan dataset *mobile price classification*

3.2 Tahap berikutnya yaitu persiapan data. Pada tahap ini dilakukan pengecekan terhadap data duplikat, *missing value*, dan data *imbalance*. Tetapi pada hasilnya tidak terdapat data duplikat, *missing value*, maupun data *imbalance*. Tahap Persiapan data bisa dilihat pada gambar 03:

```
duplicat = framedata.duplicated()
print(f"\nJumlah baris duplikat: {duplicat.sum()}")

Jumlah baris duplikat: 0
```

Gambar 03. Pengecekan data duplikat

```
Jumlah missing value:
 battery_power    0
blue              0
clock_speed       0
dual_sim          0
fc                0
four_g            0
int_memory        0
m_dep             0
mobile_wt         0
n_cores           0
pc                0
px_height         0
px_width          0
ram               0
sc_h              0
sc_w              0
talk_time         0
three_g           0
touch_screen      0
wifi              0
price_range       0
dtype: int64
```

Gambar 04. Pengecekan *Missing Value*

```
print(y.value_counts())
```

```
1    500
2    500
3    500
0    500
Name: price_range, dtype: int64
```

Gambar 05. Pengecekan data *Imbalance*

Dari hasil pengecekan persiapan data hasil menunjukkan tidak terdaoat data duplikat, *missing value* dan data dinyatakan *balanced.*

3.3 Tahap selanjutnya adalah pemodelan (*modeling*), pada tahap ini data *training* diklasifikasikan menggunakan algoritma *naive bayes* dan *K-Nearest Neighbor*. Hasil pada pemodelan ini akan dianalisis algoritma yang memiliki nilai akurasi paling tinggi dinilai lebih efektif untuk melakukan prediksi terhadap dataset. Pemodelan data dapat dilihat pada gambar 06:

```
In [168]: X = framedata.drop('price_range', axis=1)
In [169]: X
```

power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	pc	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_screen	wifi
842	0	2.2	0	1	0	7	0.6	188	2	2	20	756	2549	9	7	19	0	0	1
1021	1	0.5	1	0	1	53	0.7	136	3	6	905	1988	2631	17	3	7	1	1	0
563	1	0.5	1	2	1	41	0.9	145	5	6	1263	1716	2603	11	2	9	1	1	0
615	1	2.5	0	0	0	10	0.8	131	6	9	1216	1786	2769	16	8	11	1	0	0

Gambar 06 Pemisahan label dengan atribut

```
y = framedata['price_range']
```

```
y
```

```
0       1
1       2
2       2
3       2
4       1
       ..
1995    0
1996    2
1997    3
1998    0
1999    3
Name: price_range, Length: 2000, dtype: int64
```

Gambar 09. Menampilkan data label

Sebelum memproses dataset menggunakan algoritma, terlebih dahulu untuk memisahkan label dengan atribut lainnya. Bisa dilihat pada gambar 08 proses menghilangkan label dari atribut lainnya. Gambar 09 menampilkan label yang telah dipisah.

3.3.1. Naïve Bayes

Berikut merupakan pembahasan hasil klasifikasi menggunakan algoritma *naïve bayes*. Sebelumnya, melakukan *split* data untuk menentukan data *test* dan data *train.* Penulis melakukan beberapa kali percobaan dengan mengganti angkat pada data *test*, yang berpengaruh terhadap hasil. Tabel 01 ini merupakan perbandingannya.

Test Size : Train test	**Akurasi**
0.30 : 0.70	81
0.20 : 0.80	83

Tabel 01. Perbandingan data test dengan hasil akurasi

Dari hasil perbandingan tersebut, diambil data dengan akurasi tertinggi yaitu data *test* 0.20 dengan hasil akurasi algoritma *Naïve bayes* seperti pada gambar 10.

```
report = classification_report(y_test,y_prednb)
print(report)
```

	precision	recall	f1-score	support
0	0.95	0.86	0.90	102
1	0.77	0.70	0.74	105
2	0.71	0.83	0.76	95
3	0.92	0.93	0.92	98
accuracy			0.83	400
macro avg	0.84	0.83	0.83	400
weighted avg	0.84	0.83	0.83	400

Gambar 10. Hasil akurasi algoritma *naïve bayes*

3.3.2. K-Nearest Neighbor

Berikut merupakan pembahasan hasil klasifikasi menggunakan algoritma Pada algoritma *K-Nearest Neighbor*. Pada *K-Nearest Neighbor* juga dilakukan hal yang sama dengan melakukan perbandingan terhadap data *test*. Berikut tabel 02 perbandingan nilai data test KNN dengan hasil akurasi.

Test Size : Train test	**Akurasi**
0.30 : 0.70	92
0.20 : 0.80	91

Tabel 02. Perbandingan data test dengan hasil akurasi

Dari hasil perbandingan tersebut, diambil data dengan akurasi tertinggi yaitu data 0.30 dengan hasil akurasi algoritma *K-Nearest Neighbor* seperti pada gambar berikut.

```
report = classification_report(y_test,y_predknn)
print(report)
```

	precision	recall	f1-score	support
0	0.96	0.96	0.96	102
1	0.91	0.90	0.91	105
2	0.85	0.86	0.86	95
3	0.92	0.92	0.92	98
accuracy			0.91	400
macro avg	0.91	0.91	0.91	400
weighted avg	0.91	0.91	0.91	400

Gambar 11. Hasil akurasi algoritma KNN

3.4. Evaluasi

Pada tahap ini melakukan perbandingan hasil pengujian algoritma KNN dan *Naive Bayes*. Hasil desain dari model pengukuran akurasi dapat dilihat pada Tabel 3 dibawah ini. Sedangkan hasil perbandingan nilai akurasi dengan jurnal sebelumnya ada pada Tabel 4 berikut.

	AKURASI	DATA *TEST*	DATA *TRAIN*
NAÏVE BAYES	0.81 %	0.30	0.70
	0.83%	0.20	0.80
	AKURASI	**DATA *TEST***	**DATA *TRAIN***
KNN	0.92%	0.30	0.70
	0.91%	0.20	0.80

Tabel 03. Perbandingan hasil akurasi

Untuk melakukan evaluasi dan melakukan perbaikan, penulis membandingkan hasil akurasi dengan jurnal sebelumnya. Pada jurnal sebelumnya proses perbandingan dilakukan pada *software* rapid miner. Hasil yang diperoleh penelitian kali ini berasal dari *software jupyter notebook*.

	AKURASI
	64.21 %
NAÏVE BAYES	
	AKURASI
KNN	66.69%

Tabel 04. Perbandingan hasil akurasi dari jurnal referensi

Berdasarkan tabel hasil perbandingan akurasi diatas, menunjukkan hasil bahwa algoritma *K-Nearest Neighbor* merupakan algoritma yang paling dominan terhadap algoritma *Naïve bayes*. Suatu algoritma bisa dikatakan dominan apabila nilai hasil perbandingan akurasi dengan algoritma lain lebih kecil. Sehingga berdasarkan perbandingan nilai akurasi dan evaluasi perbandingan terhadap hasil jurnal sebelumnya, dapat disimpulkan bahwa algoritma *K-Nearest Neighbor* adalah algoritma terbaik yang dapat digunakan pada studi kasus pengklasifikasian *start-up* serta hasil pada *jupyter notebook* lebih tinggi daripada hasil nilai akurasi pada rapid miner.

3.5. Deployment

Tahapan terakhir yaitu *deployment*, tahap ini digunakan untuk melakukan prediksi keberhasilan *start-up*. Untuk data hasil prediksi yang telah dilakukan bisa dilihat pada gambar 12 untuk *naïve bayes* dan gambar 13 KNN ini

```
matrix = confusion_matrix(y_test,y_prednb)
print(matrix)
```

```
[[136  20   0   0]
 [ 11 111  35   0]
 [  0  20 108  11]
 [  0   0  12 136]]
```

Gambar 12. Hasil prediksi *naïve bayes*

```
matrix = confusion_matrix(y_test,y_predknn)
print(matrix)
```

```
[[98  4  0  0]
 [ 4 95  6  0]
 [ 0  5 82  8]
 [ 0  0  8 90]]
```

Gambar 13. Hasil prediksi kNN

KESIMPULAN

Berdasarkan hasil penelitian yang dilakukan, dapat disimpulkan bahwa algoritma *K-Nearest Neighbor* (KNN) memiliki akurasi yang lebih tinggi dibandingkan dengan algoritma *Naïve Bayes* dalam memprediksi harga *smartphone* berdasarkan fitur-fitur yang tersedia. Hal ini didukung oleh perbandingan hasil akurasi yang menunjukkan bahwa KNN lebih dominan daripada *Naïve Bayes*. Oleh karena itu, algoritma KNN dapat dianggap sebagai pilihan yang lebih baik dalam kasus pengklasifikasian harga *smartphone* daripada *Naïve Bayes.*

UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih kepada UNP Kediri dan dosen yang telah memberi bimbingan dan membantu publikasi terhadap penelitian ini.

DAFTAR PUSTAKA

[1] D. P. Utomo and M. Mesran, “Analisis Komparasi Metode Klasifikasi Data Mining dan Reduksi Atribut Pada Data Set Penyakit Jantung,” JURNAL MEDIA INFORMATIKA BUDIDARMA, vol. 4, no. 2, pp. 437–444, Apr. 2020, doi: 10.30865/MIB.V4I2.2080.

[2] N. B. Putri and A. W. Wijayanto, “Analisis Komparasi Algoritma Klasifikasi Data Mining Dalam Klasifikasi Website Phishing,” *Komputika : Jurnal Sistem Komputer*, vol. 11, no. 1, pp. 59–66, Jan. 2022, doi: 10.34010/komputika.v11i1.4350.

[3] Z. Nabila, A. R. Isnain, P. Permata, and Z. Abidin, “ANALISIS DATA MINING UNTUK CLUSTERING KASUS COVID-19 DI PROVINSI LAMPUNG DENGAN ALGORITMA K-

MEANS," *Jurnal Teknologi dan Sistem Informasi*, vol. 2, no. 2, pp. 100–108, Jul. 2021, doi: 10.33365/JTSI.V2I2.868.

[4] S. Dwi *et al.*, "Analisis Sentimen Relokasi Ibukota Nusantara Menggunakan Algoritma Naïve Bayes dan KNN," *Jurnal KomtekInfo*, vol. 10, no. 1, pp. 1–7, Jan. 2023, doi: 10.35134/KOMTEKINFO.V10I1.330.

[5] A. Felicia Watratan, A. B. Puspita, D. Moeis, S. Informasi, and S. Profesional Makassar, "Implementasi Algoritma Naive Bayes Untuk Memprediksi Tingkat Penyebaran Covid-19 Di Indonesia," *Journal of Applied Computer Science and Technology*, vol. 1, no. 1, pp. 7–14, Jul. 2020, doi: 10.52158/JACOST.V1I1.9.

[6] I. Arfanda, W. Ramdhan, R. A. Yusda, and H. Artikel, "Naive Bayes Dalam Menentukan Penerima Bantuan Langsung Tunai," *Digital Transformation Technology*, vol. 1, no. 1, pp. 9–16, Jun. 2021, doi: 10.47709/DIGITECH.V1I1.1091.

[7] N. A. Susanti, M. Walid, and H. Hoiriyah, "KLASIFIKASI DATA TWEET UJARAN KEBENCIAN DI MEDIA SOSIAL MENGGUNAKAN NAIVE BAYES CLASSIFIER," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 6, no. 2, pp. 538–543, Aug. 2022, doi: 10.36040/JATI.V6I2.5174.

[8] T. Asih, Q. Putri, A. Triayudi, and R. T. Aldisa, "Implementasi Algoritma Decision Tree dan Naïve Bayes Untuk Klasifikasi Sentimen Terhadap Kepuasan Pelanggan Starbucks," *Journal of Information System Research (JOSH)*, vol. 4, no. 2, pp. 641–649, Jan. 2023, doi: 10.47065/JOSH.V4I2.2949.

[9] A. Felicia Watratan, A. B. Puspita, D. Moeis, S. Informasi, and S. Profesional Makassar, "Implementasi Algoritma Naive Bayes Untuk Memprediksi Tingkat Penyebaran Covid-19 Di Indonesia," *Journal of Applied Computer Science and Technology*, vol. 1, no. 1, pp. 7–14, Jul. 2020, doi: 10.52158/JACOST.V1I1.9.

[10] M. Afriansyah, J. Saputra, V. Yoga Pudya Ardhana, Y. Sa, and U. Qamarul Huda Badaruddin, "ALGORITMA NAIVE BAYES YANG EFISIEN UNTUK KLASIFIKASI BUAH PISANG RAJA BERDASARKAN FITUR WARNA," *Journal of Information Systems Management and Digital Business*, vol. 1, no. 2, pp. 236–248, Jan. 2024, doi: 10.59407/JISMDB.V1I2.438.

[11] J. Homepage, Q. A'yuniyah, and M. Reza, "Penerapan Algoritma K-Nearest Neighbor Untuk Klasifikasi Jurusan Siswa Di Sma Negeri 15 Pekanbaru," *Indonesian Journal of Informatic Research and Software Engineering (IJIRSE)*, vol. 3, no. 1, pp. 39–45, Mar. 2023, doi: 10.57152/IJIRSE.V3I1.484.

[12] N. H. Purnomo, B. Pamungkas, and C. Juliane, "Penerapan Algoritma C4.5 Untuk Klasifikasi Tren Pelanggaran Kendaraan Angkutan Barang dengan Metode CRISP-DM," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 7, no. 1, pp. 30–40, Jan. 2023, doi: 10.30865/MIB.V7I1.5247.

[13] D. Kurniawan and D. M. Yasir, "Optimization Sentimen Analysis using CRISP-DM and Naive Bayes Methods Implemented on Social Media," *Cyberspace: Jurnal Pendidikan Teknologi Informasi*, vol. 6, no. 2, pp. 74–85, Oct. 2022, doi: 10.22373/CJ.V6I2.12793.

[14] N. Cholifah Sastya and D. I. Nugraha, "Penerapan Metode CRISP-DM dalam Menganalisis Data untuk Menentukan Customer Behavior di MeatSolution," *Unistek: Jurnal Pendidikan dan Aplikasi Industri*, vol. 10, no. 2, pp. 103–115, Oct. 2023, doi: 10.33592/UNISTEK.V10I2.3079.

[15] N. Widiawati, B. N. Sari, and T. N. Padilah, "Clustering Data Penduduk Miskin Dampak Covid-19 Menggunakan Algoritma K-Medoids," *Journal of Applied Informatics and Computing*, vol. 6, no. 1, pp. 55–63, May 2022, doi: 10.30871/JAIC.V6I1.3266.