

Clustering Data Stok Penjualan Sparepart Mobil Toyota Bengkel Multi Topindo Menggunakan K-Means (Clustering Sales Stock Data of Toyota Car Spare Parts Multi Topindo Workshop Using K-Means)

Yusril Haris^{1*}, John Friadi², Aurora Elsa Shafira Frederick³, Dwi Nurul Huda⁴, Mochammad Rizki Romdoni⁵

Sekolah Tinggi Teknologi Indonesia Tanjung Pinang, Kepulauan Riau^{1,3,4,5}, Universitas Batam, Batam²

yusrilharis75@gmail.com^{1*}, john.friadi@gmail.com², elsa@sttindonesia.ac.id³,
dwi@sttindonesia.ac.id⁴, rizki@sttindonesia.ac.id⁵



Riwayat Artikel

Diterima pada 28 Juni 2024

Revisi 1 pada 10 Juli 2024

Revisi 2 pada 23 Juli 2024

Disetujui pada 25 Juli 2024

Abstract

Purpose: Sales of car spare parts at the Multi Topindo Tanjungpinang Workshop play a vital role in maintaining customer satisfaction in after-sales services. Accurate data analysis is required to increase marketing effectiveness, inventory management, and customer satisfaction. One relevant approach is the application of the K-Means Clustering algorithm, which effectively groups car spare part stock data based on the initial stock and final stock. Using this method, workshops can segment stock inventory based on customer preferences by observing what spare parts are sold frequently by comparing them with the initial stock. This segmentation provides the basis for developing a more efficient sales process and precise stock management.

Methodology/approach: The research was carried out at the Multi Topindo Tanjungpinang Workshop with system development developed based on the Rapid Application Development (RAD) methodology, which consists of four stages: identifying goals and information needs, working with users to design the system, building the system, introducing a new system, and carrying out classification. spare parts sales stock data by using the k-means method.

Results/findings: The results of segmentation obtained through K-Means Clustering will help in identifying spare part groups based on the clusters that are built. In addition, this algorithm plays a role in managing inventory stock, minimizing the risk of excess or shortage of inventory, and increasing overall operational efficiency.

Limitations: The system was developed based on the K-Means Algorithm with Python programming language and Django web framework as a sample of data using a car spare parts sales inventory.

Contribution: This research contributes to the Multi-Topindo Tanjungpinang workshop, which can take more appropriate steps to meet customer demand, optimize inventory, and ultimately increase customer satisfaction.

Keywords: Clustering, Sales Stock, Sparepart, Data Mining, K-Means.

How to cite: Haris, Y., Friadi, J., Frederick, A, E, S., Huda, D, N., Romdoni, M, R. (2024). Clustering Data Stok Penjualan Sparepart Mobil Toyota Bengkel Multi Topindo Menggunakan K-Means. *Jurnal Ilmu Siber dan Teknologi Digital*, 2(2), 109-121.

1. Pendahuluan

Data science memegang peran yang sangat penting dalam berbagai aspek kehidupan dan bisnis saat ini. dalam dunia penelitian komputer, *clustering* menjadi salah satu alat penting dalam proses analisis data. Teknik ini membantu mengorganisasi data yang kompleks menjadi kelompok-kelompok yang lebih

sederhana dan terstruktur. *Data mining* adalah proses pencarian pola-pola yang menarik dan tersembunyi (*hidden pattern*) dari suatu kumpulan data yang berukuran besar yang tersimpan dalam suatu basis data (Aulia, 2021).

Data mining sendiri merupakan *sub domain* dalam *data science* yang fokus pada penemuan pola, informasi berharga, atau pengetahuan tersembunyi dalam *dataset* yang besar dan kompleks. *Data mining* juga merupakan suatu metode pengolahan data yang digunakan untuk menemukan pola tersembunyi dari data yang akan diolah (Alkhairi & Windarto, 2019). Salah satu metode yang dapat digunakan untuk melakukan analisis tersebut adalah algoritma *K-Means Clustering*. Algoritma ini merupakan metode pengelompokan data yang populer dalam menganalisis data. Tujuan dari algoritma *K-Means Clustering* adalah mengelompokkan objek data ke dalam beberapa kelompok berdasarkan kemiripan karakteristiknya. Dalam manajemen stok *spare part* mobil, algoritma ini dapat diterapkan untuk melakukan segmentasi data stok *spare part* berdasarkan pola penjualan yang terjadi. Dengan menerapkan algoritma *K-Means*, Bengkel Multi Topindo Tanjungpinang akan dapat memperoleh informasi yang berharga mengenai pola penggunaan *spare part mobil* yang berpengaruh terhadap ketersediaan stok *spare part* itu sendiri. *K-Means* juga akan memungkinkan Bengkel Multi Topindo Tanjungpinang untuk mengidentifikasi kelompok-kelompok *spare part* dengan ciri-ciri serupa. Dengan adanya pengelompokan *spare part* bengkel akan mendapatkan informasi yang bisa digunakan untuk menyusun strategi pemasaran yang lebih tepat sasaran, seperti mengirimkan penawaran khusus kepada pelanggan dalam kelompok tertentu atau mengembangkan program loyalitas yang sesuai dengan kebutuhan mereka.

Dengan mengetahui kelompok-kelompok stok *spare part* yang terbentuk, bengkel dapat melakukan estimasi kebutuhan persediaan *spare part* untuk setiap kelompok tersebut. Hal ini memungkinkan bengkel untuk mengoptimalkan stok persediaan, menghindari kelebihan atau kekurangan persediaan, dan meningkatkan penjualan *spare part*. Untuk melakukan proses *clustering* tentu membutuhkan sebuah data yang cukup banyak dan sesuai dengan yang dibutuhkan, di dalam penelitian ini peneliti menggunakan data penjualan *spare part* mobil selama tahun 2023 pada Bengkel Multi Topindo Tanjungpinang.

2. Tinjauan Pustaka dan Pengembangan Hipotesis

Tahapan dalam melakukan analisis menggunakan K-Means yaitu seleksi data, *pre-processing*, dan *data mining* (Dewi et al., 2022). Pada tahap seleksi data atau proses *knowledge discovery from database* (KDD), merupakan proses kegiatan yang melibatkan hasil dari proses *data mining* (proses mengekstrak kecenderungan pola suatu data), kemudian mengubah hasilnya secara akurat menjadi informasi yang mudah dipahami (Utomo et al., 2023). Tahap *pre-processing*, tahap dimana data yang sudah didapatkan, dipilah, dan dipisahkan agar mendapatkan data yang dibutuhkan untuk proses selanjutnya. Tahapan ini mempunyai beberapa proses dimana setiap proses tersebut saling berhubungan satu sama lainnya. Adapun proses dalam tahapan *pre-processing* adalah sebagai berikut:

a. *Data Reduction*

Proses untuk mereduksi atau mengurangi dimensi, atribut, ataupun sejumlah data yang tidak dibutuhkan dalam suatu file data (Supoyo & Prasetyaningrum, 2022).

b. *Data Cleaning*

Proses dalam tahapan *pre-processing* untuk mengisi data kosong atau blank apabila memungkinkan, duplikasi data, memperbaiki data yang tidak sesuai dengan ketentuan (F. A. Pratama et al., 2019).

c. *Data Transformation*

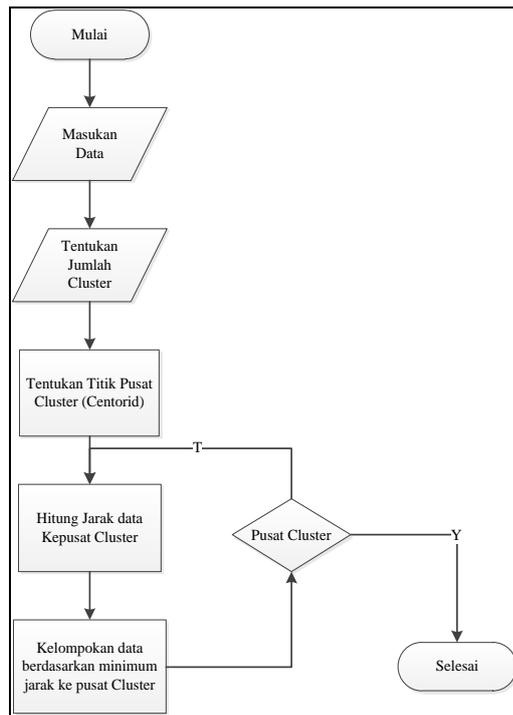
Tahap untuk mengubah data atribut yang non-numerik ke dalam nilai numerik agar data tersebut dapat diolah menggunakan algoritma *K-Means Clustering* (Siahaan, 2022).

d. *Data Integration*

Suatu proses untuk menggabungkan atau mengintegrasikan data dari beberapa file sumber.

2.1 *Data Mining*

Pada tahap ini dilakukan *clustering* dengan menggunakan algoritma K-Means yang memiliki langkah-langkah seperti pada *flowchart* berikut (gambar 1).



Gambar 1. Flowchart Algoritma K-Means

Algoritma ini dimulai dari menginputkan data stok *spare part*, yang langkah selanjutnya sebagai berikut:

1) Menentukan Jumlah *Cluster*

Cluster adalah kelompok atau grup dari data yang memiliki karakteristik yang serupa atau mirip di dalam sebuah *dataset*.

2) Menentukan *Centroid*

proses ini adalah untuk menentukan titik pusat sebagai patokan awal untuk mencari jarak antara data ke *cluster* yang sudah ditentukan.

3) Menghitung Jarak Data ke Titik Pusat *Cluster*

menghitung jarak antara data ke setiap *cluster* yang sudah dibentuk. Pada Penelitian ini menggunakan rumus *euclidean distance* (Richia Putri et al., 2023), untuk mencari jarak (*distance*) dari satu *cluster* ke *cluster* lainnya.

$$D_{(a,b)} = \sqrt{(x_{1a} - x_{1b})^2 + \dots + (x_{ka} - x_{kb})^2}$$

Dimana:

$D_{(a,b)}$: Jarak data ke a ke *centroid* b

x_{ka} : Jarak ke a pada parameter data ke k

x_{kb} : *Centroid* ke b pada parameter ke k

4) Setelah mendapatkan jarak antara setiap data ke setiap *cluster* yang sudah terbentuk, selanjutnya mengelompokan data ke dalam *cluster* berdasarkan jarak minimum (jarak terdekat) ke dalam *cluster*.

5) Menentukan *centroid* baru

Untuk menentukan *centroid* baru menggunakan rumus(2) (Dinata et al., 2020) berikut:

$$\mu_j(t + 1) = \frac{1}{N_{sj}} \sum_{j \in s_j} x_j$$

Keterangan:

$\mu_j(t + 1)$ *centroid* baru pada iterasi $(t + 1)$

N_{sj} = Data pada *cluster* S_j

6) Verifikasi *Centroid Cluster*

Apabila hasil *centroid* baru yang didapat sama dengan *centroid cluster* yang lama, maka proses K-Means sudah selesai atau disebut dengan konvergen dan hasil dari proses K-Means *clustering* sudah didapatkan.

2.2 Perhitungan Menggunakan K-Means Clustering

Pada sub-bab ini disajikan sebuah representasi perhitungan yang digunakan dalam sistem yang akan dibangun. Diawali dari Tabel 1, yang merupakan *sample* data stok yang diperoleh dari Bengkel Multi Topindo Tanjungpinang.

Tabel 1. Stok *Spare Part*

| NO | Nama Spare Part | Stok Awal | Stok Akhir |
|----|-----------------------|-----------|------------|
| 1 | Oli Mesin TMO | 10 | 6 |
| 2 | Oli Transmisi ATF | 15 | 10 |
| 3 | Oli Power Steering | 12 | 10 |
| 4 | Oli Gardan Toyota | 15 | 6 |
| 5 | Air Radiator | 20 | 2 |
| 6 | Kampas Rem Depan | 10 | 2 |
| 7 | Kampas Rem Belakang | 15 | 5 |
| 8 | Bearing Roda Depan | 10 | 8 |
| 9 | Bearing Roda Belakang | 20 | 4 |
| 10 | Baterai | 15 | 2 |
| 11 | Busi | 20 | 6 |
| 12 | Belting Dinamo | 25 | 2 |
| 13 | Filter Minyak | 20 | 5 |
| 14 | Filter Oli | 25 | 3 |
| 15 | Filter Udara | 10 | 2 |

Langkah berikutnya adalah menentukan *centroid 1* atau *centroid* awal, untuk *centroid 1* dipilih secara acak atau dapat juga dilakukan dengan mengambil sampel pada data stok sebanyak jumlah *cluster* yang akan diproses. Pada penelitian ini akan dibuat menjadi 3 *cluster* dengan *centroid* seperti pada tabel 2.

Tabel 2. Centroid 1

| No | Cluster | Stok Awal | Stok Akhir |
|----|-----------|-----------|------------|
| 1 | Cluster 1 | 10 | 3 |
| 2 | Cluster 2 | 12 | 5 |
| 3 | Cluster 3 | 14 | 8 |

Untuk menghitung jarak data ke titik pusat *cluster* menggunakan data dan *centroid cluster* yang telah ditentukan di atas, berikut adalah tiga sampel data untuk menghitung jarak menggunakan *euclidean* menggunakan rumus 1 :

a. Data: Oli Mesin TMO (stok awal: 10, stok akhir: 6)
 Jarak ke *cluster 1*:
 $\sqrt{(10 - 10)^2 + (6 - 3)^2} = 3$
 Jarak ke *cluster 2*:
 $\sqrt{(10 - 12)^2 + (6 - 5)^2} = 2,24$
 Jarak ke *cluster 3*:
 $\sqrt{(10 - 14)^2 + (6 - 8)^2} = 4,47$
 Cluser terdekat : **cluster 1**

b. Data: Oli Transmisi ATF (Stock Awal: 15, Stok Akhir: 10)
 Jarak ke *cluster 1*:
 $\sqrt{(15 - 10)^2 + (10 - 3)^2} = 8,6$
 Jarak ke *Cluster 2*:
 $\sqrt{(15 - 12)^2 + (10 - 5)^2} = 5,83$
 Jarak ke *Cluster 3*:
 $\sqrt{(15 - 14)^2 + (10 - 8)^2} = 2,24$
 Cluster Terdekat: **cluster 2**

- c. Data: Oli Power Steering (stock awal: 12, stok akhir: 10)
- Jarak ke *cluster* 1:

$$\sqrt{(12 - 10)^2 + (10 - 3)^2} = 7,28$$
- Jarak ke *Cluster* 2:

$$\sqrt{(12 - 12)^2 + (10 - 5)^2} = 5$$
- Jarak ke *cluster* 3:

$$\sqrt{(12 - 14)^2 + (10 - 8)^2} = 2,83$$
- Cluster* Terdekat: ***cluster* 3**

Lakukan perhitungan serupa untuk data-data lainnya maka akan menghasilkan data seperti tabel 3.

Tabel 3. Iterasi 1.

| NO | Cluster 1 | Cluster 2 | Cluster 3 | Cluster |
|----|-----------|-----------|-----------|---------|
| 1 | 3,00 | 2,24 | 4,47 | 2 |
| 2 | 8,60 | 5,83 | 2,24 | 3 |
| 3 | 7,28 | 5,00 | 2,83 | 3 |
| 4 | 5,83 | 3,16 | 2,24 | 3 |
| 5 | 10,05 | 8,54 | 8,49 | 3 |
| 6 | 1,00 | 3,61 | 7,21 | 1 |
| 7 | 5,39 | 3,00 | 3,16 | 2 |
| 8 | 5,00 | 3,61 | 4,00 | 2 |
| 9 | 10,05 | 8,06 | 7,21 | 3 |
| 10 | 5,10 | 4,24 | 6,08 | 2 |
| 11 | 10,44 | 8,06 | 6,32 | 3 |
| 12 | 15,03 | 13,34 | 12,53 | 3 |
| 13 | 10,20 | 8,00 | 6,71 | 3 |
| 14 | 15,00 | 13,15 | 12,08 | 3 |
| 15 | 1,00 | 3,61 | 7,21 | 1 |

Untuk menentukan apakah iterasi K-Means sudah konvergen atau belum, perlu memeriksa apakah ada perubahan dalam penempatan data ke dalam *cluster* pada iterasi terakhir. Jika pada iterasi terakhir tidak ada perubahan (data tetap berada pada *cluster* yang sama), maka iterasi sudah konvergen. Namun, jika ada perubahan, perlu melanjutkan iterasi untuk mencari *centroid cluster* baru dan melakukan perhitungan jarak kembali.

Dari tabel 1 hasil perhitungan jarak terdekat di atas, dapat dilakukan dengan cara melihat data dan *cluster* terdekat pada iterasi pertama. Selanjutnya, hitung kembali *centroid cluster* baru berdasarkan data yang sudah dikelompokkan ke dalam *cluster*.

Untuk melanjutkan iterasi, lakukan langkah-langkah berikut:

- 1) Hitung *centroid* baru untuk setiap *cluster* dengan mengambil rata-rata dari koordinat data yang berada dalam *cluster* tersebut.
- 2) Hitung kembali jarak data ke *centroid cluster* baru.
- 3) Tentukan *cluster* terdekat untuk setiap data berdasarkan jarak terkecil.
- 4) Ulangi langkah 1-3 sampai tidak ada perubahan *cluster* pada iterasi selanjutnya.

Pada iterasi selanjutnya belum memiliki data *centroid* yang dihasilkan dari iterasi sebelumnya, jadi tidak dapat melanjutkan iterasi berikutnya. Jadi, jika ingin melanjutkan iterasi K-Means, perlu melakukan langkah-langkah di atas dan mencari *centroid* baru. Berikut adalah cara untuk menentukan *centroid* baru.

Untuk menghitung *centroid* baru pada setiap *cluster* dapat dilakukan dengan menggunakan rumus 2 sebagai berikut:

| | |
|---|--|
| <p>a. <i>Centroid</i> baru <i>cluster</i> 1: Stok awal: $(10 + 10) / 2 = 10$ Stock Akhir: $(2 + 2) / 4 = 2$</p> | <p>b. <i>Centroid</i> baru untuk <i>Cluster</i> 2: Stol awal: $(10+15+10+15) / 4 = 12,50$ Stok akhir: $(6 + 5 + 8 + 2) / 4 = 5,25$</p> |
| <p>c. <i>Centroid</i> baru untuk <i>cluster</i> 3: Stok awal: $(15+12+15+20+20+20+25+20+25) / 9 = 19,11$ Stok akhir: $(10+10+6+2+4+6+2+5+3) / 9 = 5,33$</p> | |

Sehingga sekarang *centroid* barunya didapatkan seperti pada tabel 4 berikut:

Tabel 4. *Centroid* 3

| No | Cluster | Stok Awal | Stok Akhir |
|----|-----------|-----------|------------|
| 1 | Cluster 1 | 10 | 3 |
| 2 | Cluster 2 | 12.50 | 5.25 |
| 3 | Cluster 3 | 19.11 | 5.33 |

Hitung kembali jarak dari *centroid* baru di atas dengan cara yang sama sehingga akan menghasilkan data seperti pada tabel 5 berikut:

Tabel 5. Iterasi 2

| NO | Cluster 1 | Cluster 2 | Cluster 3 | Cluster |
|----|-----------|-----------|-----------|---------|
| 1 | 4,00 | 2,61 | 9,14 | 2 |
| 2 | 9,43 | 5,37 | 6,22 | 2 |
| 3 | 8,25 | 4,78 | 8,51 | 2 |
| 4 | 6,40 | 2,61 | 4,16 | 2 |
| 5 | 10,00 | 8,17 | 3,45 | 3 |
| 6 | 0,00 | 4,10 | 9,70 | 1 |
| 7 | 5,83 | 2,51 | 4,12 | 2 |
| 8 | 6,00 | 3,72 | 9,49 | 2 |
| 9 | 10,20 | 7,60 | 1,60 | 3 |
| 10 | 5,00 | 4,10 | 5,29 | 2 |
| 11 | 10,77 | 7,54 | 1,11 | 3 |
| 12 | 15,00 | 12,92 | 6,77 | 3 |
| 13 | 10,44 | 7,50 | 0,95 | 3 |
| 14 | 15,03 | 12,70 | 6,33 | 3 |
| 15 | 0,00 | 4,10 | 9,70 | 1 |

Jika diperhatikan masih ada *cluster* yang berpindah, sehingga harus melakukan iterasi berikutnya agar hasil iterasi menjadi konvergen atau tidak berpindah dari *cluster* sebelumnya ke *cluster* lain. Untuk menghitung *centroid* baru untuk setiap *cluster*, ambil rata-rata dari koordinat data yang berada dalam *cluster* tersebut.

| | |
|--|---|
| <p>a. <i>Centroid</i> baru <i>cluster</i> 1: Stok awal: $(10 + 10) / 2 = 10$ Stock Akhir: $(2 + 2) / 2 = 2$</p> | <p>b. <i>Centroid</i> baru untuk <i>Cluster</i> 2: Stol awal: $(10+15+12+15+15+10+15) / 7 = 13,14$ Stok akhir: $(6+10+10+6+5+8+2) / 7 = 6,71$</p> |
| <p>c. <i>Centroid</i> baru untuk <i>cluster</i> 3: Stok awal: $(20+20 +20+25+20+25) / 6 = 21,67$ Stok akhir: $(2+ 4 + 6+2+5+3) / 6 = 3,67$</p> | |

Sehingga sekarang *centroid* barunya didapatkan seperti pada tabel 6 berikut:

Tabel 6. *Centroid 3*

| No | Cluster | Stok Awal | Stok Akhir |
|----|-----------|-----------|------------|
| 1 | Cluster 1 | 10 | 2 |
| 2 | Cluster 2 | 13.40 | 6.71 |
| 3 | Cluster 3 | 21.67 | 3.67 |

Hitung kembali jarak dari *centroid* baru di atas dengan cara yang sama sehingga akan menghasilkan data seperti pada tabel 7 berikut:

Tabel 7. Iterasi 3

| NO | Cluster 1 | Cluster 2 | Cluster 3 | Cluster |
|----|-----------|-----------|-----------|---------|
| 1 | 4,00 | 3,22 | 11,90 | 2 |
| 2 | 9,43 | 3,77 | 9,20 | 2 |
| 3 | 8,25 | 3,48 | 11,56 | 2 |
| 4 | 6,40 | 1,99 | 7,06 | 2 |
| 5 | 10,00 | 8,32 | 2,36 | 3 |
| 6 | 0,00 | 5,67 | 11,79 | 1 |
| 7 | 5,83 | 2,53 | 6,80 | 2 |
| 8 | 6,00 | 3,40 | 12,45 | 2 |
| 9 | 10,20 | 7,37 | 1,70 | 3 |
| 10 | 5,00 | 5,07 | 6,87 | 1 |
| 11 | 10,77 | 6,89 | 2,87 | 3 |
| 12 | 15,00 | 12,76 | 3,73 | 3 |
| 13 | 10,44 | 7,07 | 2,13 | 3 |
| 14 | 15,03 | 12,43 | 3,40 | 3 |
| 15 | 0,00 | 5,67 | 11,79 | 1 |

Pada iterasi di atas masih ada *cluster* yang berpindah, sehingga harus melakukan iterasi berikutnya agar hasil iterasi menjadi konvergen atau tidak berpindah dari *cluster* sebelumnya ke *cluster* lain. Untuk menghitung *centroid* baru untuk setiap *cluster*, ambil rata-rata dari koordinat data yang berada dalam *cluster* tersebut.

a. *Centroid* baru cluster 1:

Stok awal: $(10+15+10) / 3 = 11,67$

Stok Akhir: $(2+2+2) / 3 = 2$

b. *Centroid* baru untuk Cluster 2:

Stol awal: $(10 + 15 + 12 + 15 + 15+10) / 6 = 12,83$

Stok akhir: $(6+10+10 +6+5+8) / 6 = 7,5$

c. *Centroid* baru untuk cluster 3:

Stok awal: $(20+20+20+25+20+25) / 6 = 21,67$

Stok akhir: $(2+4+6+2+5+3) / 6 = 3,67$

Sehingga sekarang *centroid* barunya adalah seperti pada tabel 8.

Tabel 8. *Centroid 4*

| No | Cluster | Stok Awal | Stok Akhir |
|----|-----------|-----------|------------|
| 1 | Cluster 1 | 11.67 | 2 |
| 2 | Cluster 2 | 12.83 | 7.5 |
| 3 | Cluster 3 | 21.67 | 3.67 |

Hitung kembali jarak dari *centroid* baru di atas dengan cara yang sama sehingga akan menghasilkan data seperti pada tabel 9 berikut :

Tabel 9. Iterasi 4

| NO | Cluster 1 | Cluster 2 | Cluster 3 | Cluster |
|----|-----------|-----------|-----------|---------|
| 1 | 4,33 | 3,21 | 11,90 | 2 |
| 2 | 8,67 | 3,31 | 9,20 | 2 |
| 3 | 8,01 | 2,64 | 11,56 | 2 |
| 4 | 5,21 | 2,64 | 7,06 | 2 |
| 5 | 8,33 | 9,03 | 2,36 | 3 |
| 6 | 1,67 | 6,19 | 11,79 | 1 |
| 7 | 4,48 | 3,31 | 6,80 | 2 |
| 8 | 6,23 | 2,88 | 12,45 | 2 |
| 9 | 8,57 | 7,98 | 1,70 | 3 |
| 10 | 3,33 | 5,91 | 6,87 | 1 |
| 11 | 9,24 | 7,32 | 2,87 | 3 |
| 12 | 13,33 | 13,35 | 3,73 | 3 |
| 13 | 8,86 | 7,59 | 2,13 | 3 |
| 14 | 13,37 | 12,97 | 3,40 | 3 |
| 15 | 1,67 | 6,19 | 11,79 | 1 |

Pada tabel di atas sudah tidak ada *cluster* yang berpindah sehingga data sudah konvergen dan tidak perlu lagi dilakukan iterasi. Setelah data konvergen selanjutnya kelompokkan lagi data tersebut untuk melihat anggota pada setiap *cluster*. Berikut adalah hasil akhir pada *clustering* data penjualan *sparepart* pada bengkel Multi Topindo Tanjungpinang :

Tabel 10. Hasil Akhir Proses *Clustering*

| No | Cluster | Kelompok |
|----|-----------|-----------------|
| 1 | Cluster 1 | 6,10,15 |
| 2 | Cluster 2 | 1,2,3,4,7,8 |
| 3 | Cluster 3 | 5,9,11,12,13,14 |

Data *spare part* telah dikelompokkan berdasarkan kesamaan jaraknya dengan *centroid* pada masing-masing *cluster*.

a. Cluster 1 beranggotakan:

Kampas Rem Depan, Baterai, Filter Udara

b. Cluster 2 beranggotakan:

Oli Mesin TMO, Oli Trasmisi ATF, Oli Power Steering, Oli Gardan, Kampas Rem Belakang, Bearing Roda Depan

c. Cluster 3 beranggotakan:

Air Radiator, Bearing Roda Belakang, Busi, Belting Dinamo, Filter Minyak, Filter Oli

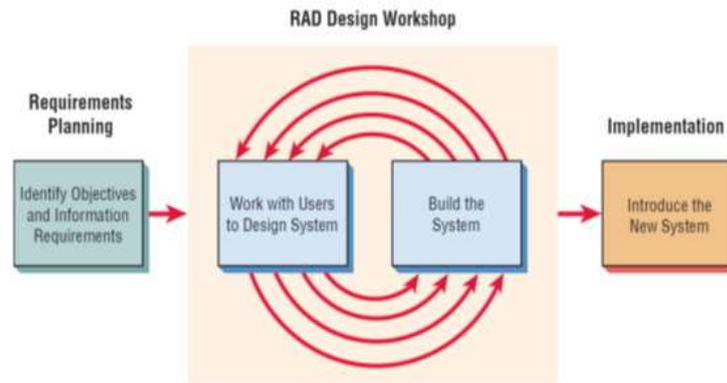
Penamaan kelompok dapat digunakan berdasarkan rata – rata *centroid* akhir dari setiap *cluster* berikut adalah penamaan untuk setiap *cluster*:

Tabel 11. Hasil Akhir Proses *Clustering*

| No | Cluster | Nama Cluster | Stok Awal | Stok Akhir |
|----|-----------|------------------------------|-----------|------------|
| 1 | Cluster 1 | Stok Rendah Penjualan Tinggi | 11,67 | 2 |
| 2 | Cluster 2 | Stok Sedang Penjualan Rendah | 12,83 | 7,5 |
| 3 | Cluster 3 | Stok Tinggi Penjualan Sedang | 21,67 | 3,67 |

3. Metodologi Penelitian

Bagian ini menjelaskan materi, metode, survey, kuesioner, dll. yang digunakan dalam penelitian. jelaskan apakah penelitian yang digunakan adalah *experimental*, *review study*, *simulation based*, atau berdasarkan survey. Deskripsikan software dan hardware yang digunakan dalam penelitian beserta merknya. Sebutkan semua riset penelitian, asumsi disertai teori pendukung. Bagian ini harus gamblang sehingga memudahkan pembaca untuk mengulangi penelitian dengan kondisi yang mirip.



Gambar 2. Metode *Rapid Application Development*

Adapun fase-fase model pengembangan sistem *Rapid Application Development* (RAD) sebagai berikut (F. D. Pratama et al., 2022):

- a. Perencanaan Kebutuhan (*Requirement Planning*)
Perencanaan Kebutuhan (*Requirement Planning*) adalah langkah awal dalam pengembangan sistem. Pada tahap ini, data dikumpulkan melalui berbagai metode seperti penelitian, wawancara, atau studi literatur.
- b. Tahap Desain Pengguna (*Users Design*)
Pada tahap ini dilakukan proses pemodelan dengan tujuan untuk pembuatan prototype sementara, sebagai bahan evaluasi dalam perbaikan apabila masih terdapat ketidaksesuaian antara pengguna dan sistem analisis.
- c. Tahap Membangun Sistem (*Build System*)
Tahap ini, aktivitas membangun aplikasi dilakukan sesuai dengan data yang telah diperoleh dari tahap perencanaan kebutuhan.
- d. Implementasi (*Implementation*)
Pada fase ini, penganalisis bekerja dengan para pengguna secara intens selama *workshop* dan merancang aspek-aspek bisnis dan nonteknis perusahaan. Segera setelah aspek-aspek ini disetujui dan sistem-sistem dibangun dan disaring, sistem-sistem baru atau bagian dari sistem diujicoba dan kemudian diperkenalkan kepada organisasi.

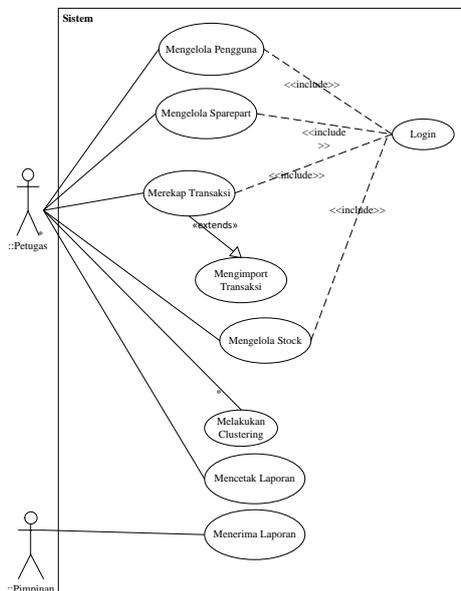
4. Hasil dan Pembahasan

Hasil dan pembahasan harus memuat analisis penemuan penelitian yang logis dan ilmiah. Sajikan bukti untuk mendukung analisis Anda dengan mengutip penelitian-penelitian atau teori sebelumnya. Jika terdapat tabel dan gambar, harap ditulis dengan format berikut: judul dan sumber tabel berada rata kiri, tabel berada di tengah, gunakan hanya garis horizontal untuk bagian luar tabel, bagian dalam tabel tidak menggunakan garis vertikal maupun horizontal.

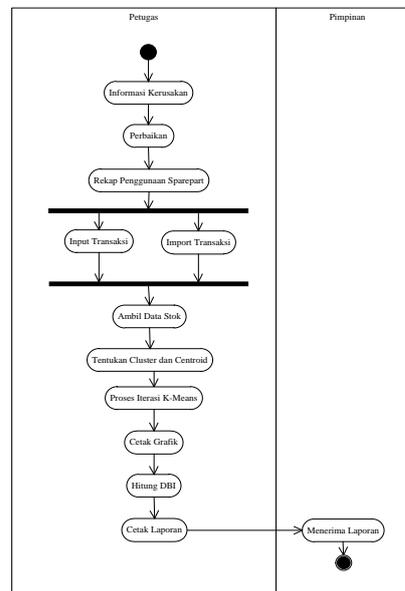
4.1 Perancangan Sistem

Entitas-entitas yang terlibat dalam sistem digambarkan melalui diagram *use case* yang merupakan alat visual untuk menggambarkan interaksi antara aktor-aktor (pengguna) dengan sistem atau aplikasi yang sedang dikembangkan (Wahyuni, 2017). Seperti pada gambar 3, ada 2 aktor yang terlibat yaitu Petugas dan Pimpinan. Petugas mengelola pengguna, mengelola *spare part*, merekap transaksi, mengimpor transaksi, mengelola stok, dan melakukan *clustering*; sedangkan Pimpinan menerima laporan dari sistem. Secara singkat dari *use case* dijelaskan secara rinci melalui diagram *activity*, suatu diagram yang

menggambarkan konsep aliran data/kontrol (Wahyuni, 2017), *Activity* diagram adalah salah satu jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan alur kerja atau aliran aktivitas dalam suatu sistem, proses, atau interaksi. Gambar 4 menggambarkan aktifitas Petugas dimulai dari rekap penggunaan *spare part*, input transaksi, ambil data stok, tentukan *cluster* dan *centroid*, proses iterasi K-Means, cetak grafik, dan hitung DBI (*Davies Bouldin Index*), dan terakhir cetak laporan yang akan diserahkan pada Pimpinan.



Gambar 3. Use Case Diagram



Gambar 4. Activity Diagram

Tahap desain suatu perangkat lunak digambarkan melalui diagram *class*. *Class* diagram terdiri dari beberap simbol diantaranya adalah Kelas, Interface, Asosiasi, Generalisasi (Apriadi et al., 2019). *Class* yang dibuat terdiri dari tiga *class* yaitu "Sparepart," "Stock," dan "Transaction," beserta atribut dan metode yang dimiliki masing-masing kelas.

1) *Sparepart Class*:

- a. Atribut : ``part_id``: ID unik untuk *sparepart*, ``part_name``: Nama *sparepart*, ``part_price``: Harga *sparepart*.
- b. Metode: ``save()``: Simpan informasi *sparepart* ke dalam basis data, ``edit()``: Edit informasi *sparepart* yang sudah ada, ``delete()``: Hapus data *sparepart* dari basis data.

2) *Stock Class*:

- a. Atribut: ``stock_id``: ID unik untuk stock, ``part_id``: ID *sparepart* yang terkait, ``initial_stock``: Stock awal, ``final_stock``: Stock akhir setelah transaksi.
- b. Metode: ``save()``: Simpan informasi stock ke dalam basis data, ``edit()``: Edit informasi stock yang sudah ada. ``delete()``: Hapus data stock dari basis data, ``clustering()``: Proses analisis *clustering* untuk stock. ``printGrafik()``: Cetak grafik visualisasi data stock, ``getDBI()``: Dapatkan nilai *Davies-Bouldin Index* dari hasil *clustering*.

3) *Transaction Class*:

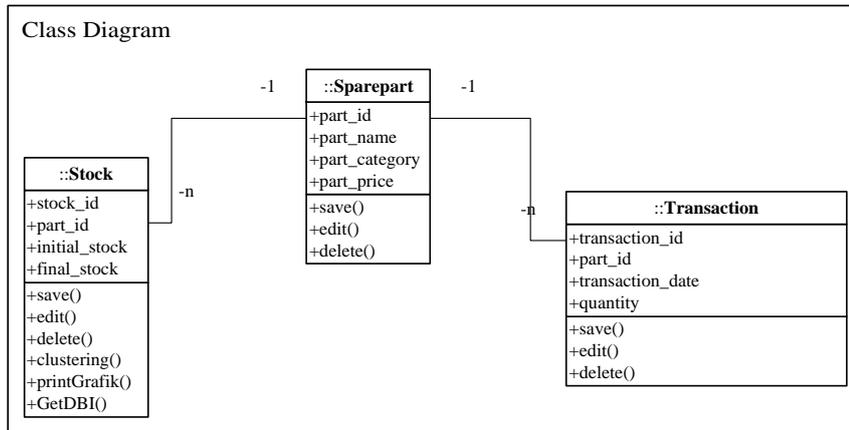
- a. Atribut: ``transaction_id``: ID unik untuk transaksi, ``part_id``: ID *sparepart* yang terlibat dalam transaksi, ``transaction_date``: Tanggal transaksi. ``quantity``: Jumlah barang yang terlibat dalam transaksi.
- b. Metode: ``save()``: Simpan informasi transaksi ke dalam basis data, ``edit()``: Edit informasi transaksi yang sudah ada, ``delete()``: Hapus data transaksi dari basis data.

4) Hubungan

Hubungan atau disebut juga asosiasi adalah hubungan antar objek yang saling membutuhkan. Sebuah kelas dapat diasosiasikan dengan beberapa kelas sekaligus.

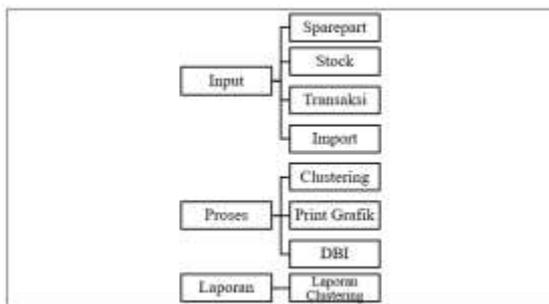
- a. Kelas "Stock" memiliki asosiasi dengan kelas "Sparepart" melalui atribut ``part_id``. Ini menunjukkan bahwa kelas "Stok" terkait dengan informasi dari kelas "Sparepart".

- b. Kelas "Transaction" juga memiliki asosiasi dengan kelas "Sparepart" melalui atribut `part_id`. Ini menunjukkan bahwa kelas "Transaction" terkait dengan informasi dari kelas "Sparepart."

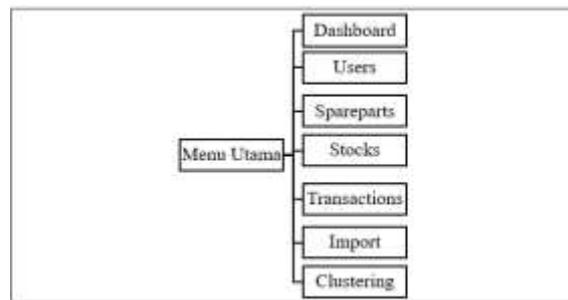


Gambar 5. Class Diagram

Sistem yang bangun memiliki struktur program seperti gambar 6. struktur program merupakan konsep pembuatan program dimana instruksi-instruksi di dalam program dikelompokkan ke dalam subprogram-subprogram sehingga setiap subprogram mempunyai tugas dan fungsi tersendiri. Struktur menu memudahkan pengguna dalam menggunakan aplikasi dengan berbagai fungsi yang ada, maka perlu dibuat sebuah struktur menu untuk mengelompokkan fasilitas-fasilitas yang disediakan aplikasi.



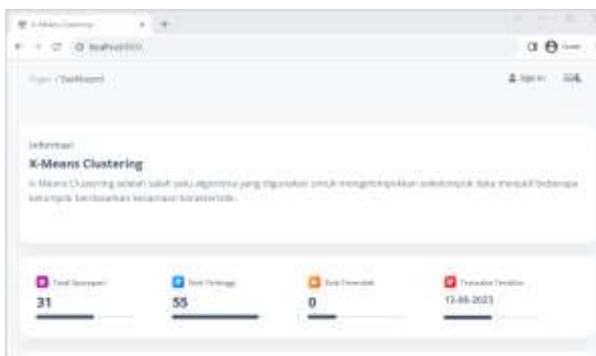
Gambar 6. Struktur Program



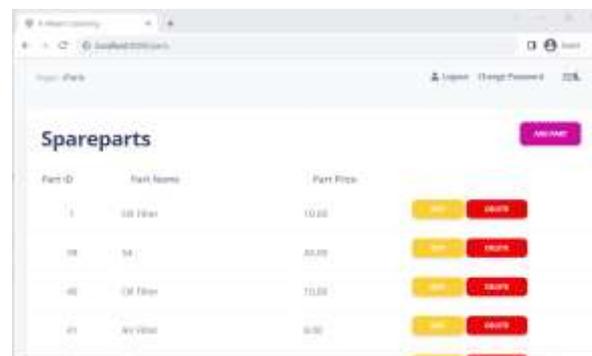
Gambar 7. Struktur Menu

4.2 Implementasi Antar Muka

Setelah tahap perancangan dilanjutkan dengan implementasi sistem. Gambar 8.a adalah halaman dashboard yang menampilkan data statistic stok. Petugas menginputkan data *spare part* pada halaman seperti gambar 8.b



Gambar 8.a Halaman Dashboard



Gambar 8.b. Halaman Sparepart

Petugas menginputkan data stok di halaman seperti gambar 10.a dan transaksi penggunaan *spare part* di halana seperti gambar 10.b

| Stock ID | Part Name | Initial Stock | Final Stock |
|----------|------------|---------------|-------------|
| 1 | Oil Filter | 5 | 15 |
| 2 | Wax | 5 | 15 |
| 3 | Oil Filter | 5 | 15 |
| 4 | Wax | 5 | 15 |

Gambar 10.a. Halaman Stok

| Transaction ID | Part Name | Transaction Date | Quantity |
|----------------|------------|------------------|----------|
| 15 | Oil Filter | Aug 5, 2020 | 7 |
| 11 | Oil Filter | Jul 11, 2020 | 4 |
| 16 | Brake Pad | Aug 1, 2020 | 2 |
| 10 | Spark Plug | Aug 6, 2020 | 8 |

Gambar 10.b. Halaman Transaksi

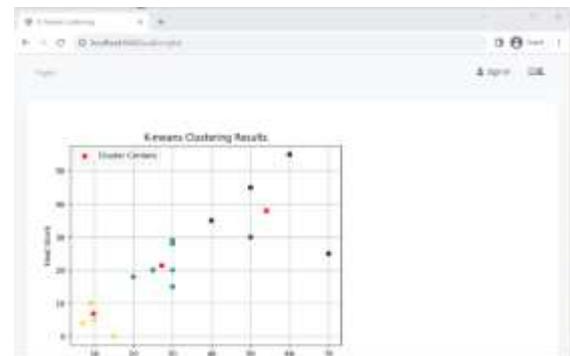
Melalui halaman impor (gambar 11.a), Petugas dapat mengisi data transaksi secara sekaligus tanpa menginput satu per satu. Sebelum proses *clustering*, Petugas perlu mengisi parameter yang diperlukan seperti jumlah *cluster*, stok awal dan stok akhir.

Gambar 11.a. Halaman Impor Data Transaksi

Gambar 11.b. Laporan Clustering

| Initial Stock | Final Stock | Cluster | Cluster Center Initial Stock | Cluster Center Final Stock |
|---------------|-------------|---------|------------------------------|----------------------------|
| 0.0 | 15.0 | 0.0 | 38.5 | 38.0 |
| 0.0 | 15.0 | 0.0 | 27.542607142857142 | 27.426871428571427 |
| 0.0 | 15.0 | 0.0 | 8.714285714285714 | 7.0000000000000002 |
| 0.0 | 15.0 | 0.0 | 14.0 | 14.0 |
| 1.0 | 4.0 | 0.0 | 10.0 | 10.0 |
| 33.0 | 5.0 | 0.0 | 30.0 | 30.0 |
| 15.0 | 5.0 | 0.0 | 14.0 | 14.0 |

Gambar 12.a Hasil K-Means



Gambar 12.b Grafik Scatter Plot

5. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan penulis menyimpulkan bahwa :

- 1) Melalui metode *clustering*, bengkel Bengkel Multi Topindo Tanjungpinang berhasil memahami pola penjualan sparepart mobil dengan lebih jelas. Dengan menggunakan metode tersebut sekarang memiliki wawasan yang lebih baik tentang tren penggunaan *sparepart*, termasuk produk-produk yang paling diminati oleh pelanggan.
- 2) Dengan menerapkan *clustering*, data stok *sparepart* dapat dielompokkan menjadi kategori-kategori yang lebih relevan dan berdasarkan karakteristik yang serupa. Hal ini membantu dalam melacak persediaan secara lebih efisien dan mengelola kebutuhan pelanggan dengan lebih baik.
- 3) Metode *clustering* memberikan landasan yang sistematis untuk mengoptimalkan pengelolaan persediaan. Bengkel Multi Topindo Tanjungpinang dapat mengidentifikasi kekurangan stock dan menghindari kelebihan stok yang tidak perlu, yang pada gilirannya meningkatkan efisiensi operasional dan profitabilitas bisnis.

Referensi

- Alkhairi, P., & Windarto, A. P. (2019). *Penerapan K-Means Cluster Pada Daerah Potensi Pertanian Karet Produktif di Sumatera Utara*.
- Apriadi, H., Amalia, F., & Priyambadha, B. (2019). Pengembangan Aplikasi Kakas Bantu Untuk Menghitung Estimasi Nilai Modifiability Dari Class Diagram. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3(11).
- Aulia, S. (2021). Klasterisasi Pola Penjualan Pestisida Menggunakan Metode K-Means Clustering (Studi Kasus Di Toko Juanda Tani Kecamatan Hutabayu Raja). *Djtechno: Jurnal Teknologi Informasi*, 1(1), 1–5. <https://doi.org/10.46576/djtechno.v1i1.964>
- Dewi, N. L. P. P., Purnama, I. N., & Utami, N. W. (2022). Penerapan Data Mining Untuk Clustering Penilaian Kinerja Dosen Menggunakan Algoritma K-Means (Studi Kasus: STMIK Primakara). *Jurnal Ilmiah Teknologi Informasi Asia*, 16(2), 105. <https://doi.org/10.32815/jitika.v16i2.761>
- Dinata, R. K., Hasdyna, N., & Azizah, N. (2020). *Analisis K-Means Clustering pada Data Sepeda Motor*. 5(1).
- Friadi, J., Yani, D. P., Zaid, M., & Sikumbang, A. (2023). Perancangan Pemodelan Unified Modeling Language Sistem Antrian Online Kunjungan Pasien Rawat Jalan Pada Puskesmas. *Jurnal Ilmu Siber dan Teknologi Digital*, 1(2), 125-133.
- Husnul, A. H., Nurhatsyah, N., & Friadi, J. (2022). Sistem Informasi Pariwisata Pantai Glory Melur Berbasis Web. *Jurnal Ilmu Siber dan Teknologi Digital*, 1(1), 53-64.
- Pratama, F. A., Narasati, R., & Amalia, D. R. (2019). *Pengaruh Kata Cashback Terhadap Peningkatan Penjualan Menggunakan Data Mining*. 3(2).
- Pratama, F. D., Zufria, I., & Triase, T. (2022). Implementasi Data Mining Menggunakan Algoritma Naïve Bayes Untuk Klasifikasi Penerima Program Indonesia Pintar. *Rabit : Jurnal Teknologi dan Sistem Informasi Univrab*, 7(1), 77–84. <https://doi.org/10.36341/rabit.v7i1.2217>
- Richia Putri, M., Gibran Satya Nugraha, & Ramaditia Dwiyanaputra. (2023). Pengelompokan Provinsi di Indonesia Berdasarkan Indikator Pendidikan Menggunakan Metode K-Means Clustering: Grouping Provinces in Indonesia Based on Education Indicators Using the K-Means Clustering Method. *Journal of Computer Science and Informatics Engineering (J-Cosine)*, 7(1). <https://doi.org/10.29303/jcosine.v7i1.509>
- Siahaan, M. (2022). Data Mining Strategi Pembangunan Infrastruktur Menggunakan Algoritma K-Means. *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, 11(3), 316–324. <https://doi.org/10.32736/sisfokom.v11i3.1453>
- Supoyo, A., & Prasetyaningrum, P. T. (2022). Analisis Data Mining Untuk Memprediksi Lama Perawatan Pasien Covid-19 Di DIY. *Bianglala Informatika*, 10(1), 21–29. <https://doi.org/10.31294/bi.v10i1.11890>
- Suroto, S., & Friadi, J. (2023). Manajemen Risiko Teknologi Informasi Pada Aplikasi CMS di PT. Sarana Citranusa Kabil-Batam Menggunakan ISO31000: 2018. *Jurnal Ilmu Siber dan Teknologi Digital*, 1(2), 135-148.
- Utomo, Y. B., Kurniasari, I., & Yanuartanti, I. (2023). Penerapan Knowledge Discovery In Database Untuk Analisa Tingkat Kecelakaan Lalu Lintas. *JTIK (Jurnal Teknik Informatika Kaputama)*, 7(1), 171–180. <https://doi.org/10.59697/jtik.v7i1.61>
- Wahyuni, W. (2017). Rekayasa Ulang (Reengineering) Sistem Informasi Manajemen Pertanahan Nasional dengan Pendekatan Unified Modelling Language (UML). *BHUMI: Jurnal Agraria dan Pertanahan*, 3(1), 111. <https://doi.org/10.31292/jb.v3i1.94>