

CESS

(Journal of Computer Engineering, System and Science)

Available online: <https://jurnal.unimed.ac.id/2012/index.php/cess>

ISSN: 2502-714x (Print) | ISSN: 2502-7131 (Online)



Penerapan Algoritma K-Means Untuk Clustering *Framework* dan *Library* PHP Berdasarkan Analisis *Code Structure*

Implementation Of K-Means Algorithm for Clustering Framework and PHP Library Based on Code Structure Analysis

Muhammad Ridho Putra Syalabi^{1*}, Ahmad Fahrudi Setiawan², Eko Heri Susanto³

^{1,2,3}Institut Teknologi Nasional Malang

Jalan Raya Karanglo km 2 Malang, Indonesia

Email: ¹2118014@scholar.itn.ac.id, ²fahrudi.itn@gmail.com, ³ekoheris@lecturer.itn.ac.id

*Corresponding Author

ABSTRAK

Framework dan *library* PHP memiliki variasi struktur kode yang signifikan, sehingga pengembang menghadapi tantangan dalam memilih tools yang sesuai untuk kebutuhan proyek. Penelitian ini bertujuan untuk mengelompokkan *framework* dan *library* PHP berdasarkan analisis struktur kode menggunakan algoritma K-Means dengan sistem berbasis *website*. Analisis dilakukan dengan mempertimbangkan 13 atribut, seperti LOC, NCLOC, CLOC, rata-rata panjang kelas, panjang metode, dan distribusi fungsi, untuk menemukan pola distribusi dalam kelompok yang signifikan. Meskipun *Elbow Method* menunjukkan $k=4$ (91.799) sebagai cluster optimal berdasarkan penurunan WCSS yang signifikan, analisis lanjutan dengan *Silhouette Score* dan evaluasi visual menunjukkan bahwa pembagian data menjadi 3 cluster memberikan pemisahan yang lebih jelas dan mudah diinterpretasikan. Cluster 1 mewakili *framework* dengan struktur kode kecil dan sederhana, sementara Cluster 2 dan Cluster 3 mencerminkan *framework* dengan struktur yang lebih kompleks dan ukuran kode yang lebih besar. Oleh karena itu, jumlah cluster 3 dipilih sebagai hasil yang optimal, karena memberikan keseimbangan antara kompleksitas model dan interpretasi yang jelas. Hasil clustering menggunakan sistem berbasis *website* menunjukkan presisi lebih tinggi, dengan perhitungan jarak yang mencapai hingga 15 angka di belakang koma, dibandingkan perhitungan manual yang hanya mencapai 5-6 angka, yang mengindikasikan peningkatan akurasi. Penelitian ini memberikan kontribusi pada pengelompokan *framework* PHP yang lebih efisien dan membantu pengembang memilih *tools* yang sesuai.

Kata Kunci: *K-Means; Struktur kode; Framework PHP; Library PHP; Silhouette Score; Kompleksitas kode; Clustering*



ABSTRACT

PHP frameworks and libraries exhibit significant variation in code structure, presenting developers with challenges in selecting the appropriate tools for their project needs. This study aims to group PHP frameworks and libraries based on code structure analysis using the K-Means algorithm with a website-based system. The analysis considers 13 attributes, such as LOC, NCLOC, CLOC, average class length, method length, and function distribution, to identify significant distribution patterns within the clusters. Although the Elbow Method indicates that $k=4$ is the optimal number of clusters based on a significant decrease in WCSS, further analysis using Silhouette Score and visual evaluation shows that dividing the data into 3 clusters provides clearer separation and is easier to interpret. Cluster 1 represents frameworks with small and simple code structures, while Clusters 2 and 3 reflect frameworks with more complex structures and larger code sizes. Therefore, a 3-cluster solution is chosen as the optimal result, as it provides a balance between model complexity and clear interpretation. Clustering results using the website-based system show higher precision, with distance calculations reaching up to 15 decimal places, compared to manual calculations, which only reach 5-6 decimal places, indicating improved accuracy. This study contributes to more efficient clustering of PHP frameworks and helps developers choose the right tools..

Keywords: *K-Means; Code structure; PHP Frameworks; PHP Libraries; Silhouette Score; Code complexity; Clustering*

1. PENDAHULUAN

Pemilihan teknologi *stack* yang tepat dalam pengembangan perangkat lunak modern sangat penting untuk keberhasilan proyek, terutama untuk aplikasi web. Teknologi *stack* terdiri dari berbagai teknologi seperti *framework*, *library*, dan alat bantu lainnya. Keputusan ini menentukan bagaimana kode dikelola, dikembangkan, dan diintegrasikan, yang berdampak pada kinerja aplikasi serta skalabilitas dan kemudahan penggunaan [1].

Meskipun framework PHP seperti CodeIgniter dan Laravel memberikan kemudahan dalam proses pengembangan melalui kerangka kerja yang telah terstruktur, masing-masing framework memiliki struktur kode, standar pemrograman, dan arsitektur yang berbeda, yang berdampak pada efisiensi pengembangan, skalabilitas, dan kemudahan pemeliharaan[2]. Sebagai contoh, penelitian oleh Wini Muthia Kansha et al. (2023) menemukan bahwa kesalahan dalam memilih framework dapat memperlambat proses pengembangan dan menyebabkan masalah dalam pengelolaan kode. Karena banyaknya framework yang tersedia, pengembang memerlukan pendekatan yang lebih terstruktur untuk memahami manfaat dan kelemahan masing-masing framework agar dapat membuat keputusan yang tepat[3].

Salah satu cara untuk membantu pengembang memahami pola dan karakteristik framework yang sesuai dengan kebutuhan proyek adalah dengan melakukan analisis struktur kode framework. Sebagaimana ditunjukkan oleh Robby Yuli Endra et al. (2021), struktur kode yang terorganisir seperti Laravel dapat meningkatkan skalabilitas aplikasi dan mempermudah manajemen modul. Namun, masih jarang dilakukan secara sistematis untuk membandingkan dan mengelompokkan framework berdasarkan struktur kode mereka[4].

Metode clustering dan pendekatan berbasis analitik sangat berpotensi untuk menyelesaikan masalah ini. Misalnya, algoritma K-Means dapat digunakan untuk menempatkan framework ke da kelompok berdasarkan seberapa mirip struktur kode mereka.

Metode eksploratif ini menawarkan wawasan eksploratif yang dapat membantu pengembang memahami pola-pola struktural rangka kerja dan mengurangi risiko saat memilih teknologi [5].

Tujuan penelitian ini adalah untuk membuat aplikasi berbasis algoritma K-Means yang menganalisis dan mengelompokkan *framework* dan *library* PHP berdasarkan fitur struktur kode. Penelitian ini tidak hanya memberikan manfaat praktis bagi pengembang perangkat lunak, tetapi juga memperluas wawasan teoritis tentang analisis dan pemilihan *framework* berbasis data. Diharapkan aplikasi ini akan memberikan pemahaman mendalam tentang pola struktural *framework*, membantu pengembang membuat pilihan teknologi yang lebih baik, dan meningkatkan efisiensi proses pengembangan aplikasi *web*.

2. TINJAUAN TEORI

2.1. Framework

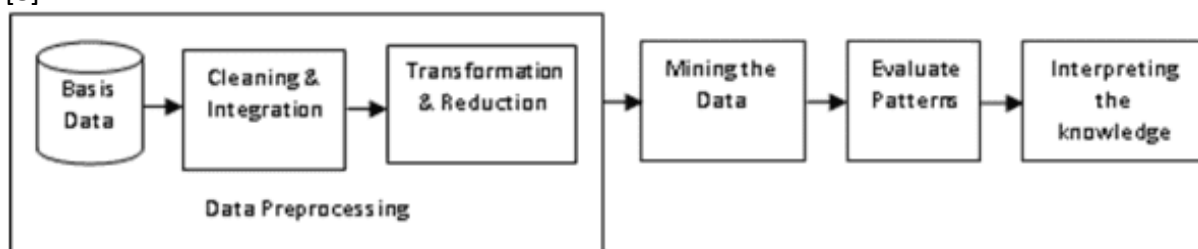
Framework adalah sekumpulan kelas, fungsi, dan aturan yang mengatur bagaimana aplikasi dibangun, memungkinkan pengembang untuk fokus pada masalah inti sementara komponen pendukung seperti koneksi database dan validasi formulir sudah disediakan oleh *framework* itu sendiri. Hal ini meningkatkan kecepatan pengembangan aplikasi [6].

2.2. Library

Library adalah kumpulan kode yang dimaksudkan untuk menyederhanakan proses pengembangan aplikasi dengan menyediakan fungsi yang dapat digunakan kembali. Dalam pengembangan web, library membantu pengembang mengimplementasikan fitur tertentu tanpa harus menulis kode dari awal, meningkatkan produktivitas dan efisiensi [7].

2.3. Data Mining

Data mining adalah proses menemukan informasi penting dari banyak basis data untuk menghasilkan wawasan baru yang membantu dalam pengambilan keputusan. Menurut Wu et al. (2021), proses data mining terdiri dari enam tahap utama. pemilihan data, pengolahan dan integrasi data, pembersihan dan transformasi data, data mining, evaluasi, dan penemuan hasil [8].



Gambar 1. Proses Data Mining

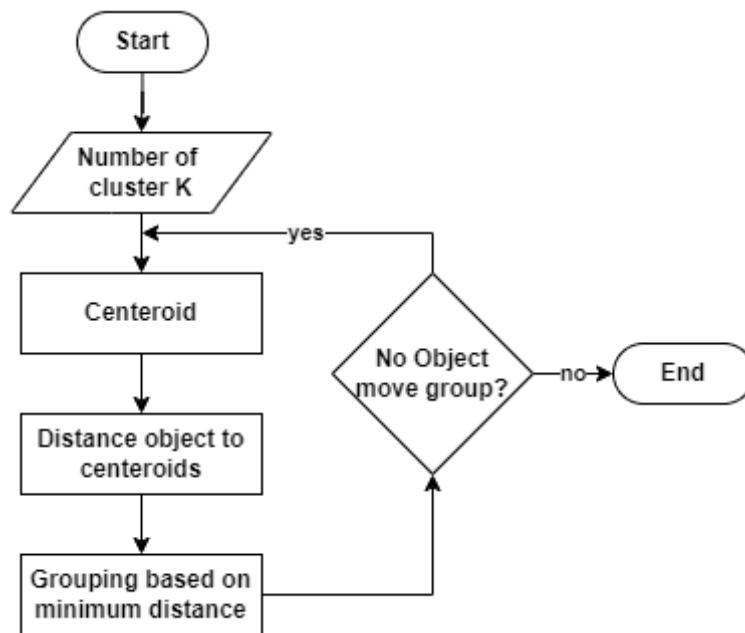
2.4. Algoritma K-Means

Metode pengelompokan data tanpa bimbingan yang dikenal sebagai metode K-Means Clustering membagi data menjadi berbagai cluster berdasarkan kesamaan karakteristik. Tujuan dari metode ini adalah untuk mengurangi variasi antar cluster dan lebih banyak variasi dalam cluster [9].

Istilah-istilah K-Means termasuk:

- 1) *Centroid* adalah titik pusat *cluster*, yang merupakan posisi rata-rata dari semua titik dalam cluster.
- 2) *Cluster* adalah kelompok data yang dikumpulkan berdasarkan seberapa dekat mereka berada dengan *centroid*.

- 3) Iterasi: Pengulangan berlangsung sampai pembagian *cluster* atau posisi *centroid* tetap [4].



Gambar 2. Flowchart Algoritma K-Means

Algoritma K-Means dimulai dengan menghitung jumlah cluster (K) dan secara acak memilih centroid awal. Untuk mengelompokkan objek berdasarkan kedekatannya dengan centroid, jarak geometris digunakan. Prosedur ini diulang hingga pembagian cluster tidak berubah, dan centroid diperbarui berdasarkan posisi rata-rata objek dalam setiap cluster [9]. Adapaun tahap dari algoritma K-Means adalah sebagai berikut [9]:

- 1) Tahap pertama Algoritma K-Means adalah menentukan jumlah cluster K yang diinginkan, yang berdampak pada hasil clustering. Nilai K ditentukan dengan menggunakan pengetahuan domain atau metode seperti Metode Elbow. Selanjutnya, K titik acak dipilih untuk menjadi centroid awal.
- 2) Kemudian, algoritma menghitung jarak geometris antara semua centroid yang telah dipilih dan setiap objek data. Jarak geometris ini dihitung dengan rumus tertentu dan mengukur jarak linier antara dua titik di ruang multidimensi.

$$d(X_i, C_j) = \sqrt{\sum_{m=1}^n (X_{im} - C_{jm})^2} \quad (1)$$

Dimana X_{im} dan C_{jm} adalah jarak antara objek data dan centroid pada dimensi m . Setelah menghitung jarak, setiap objek kemudian dikelompokkan ke dalam cluster yang memiliki centroid terdekat, yang memastikan bahwa setiap objek berada dalam cluster yang meminimalkan jarak ke centroid sehingga membentuk cluster awal.

- 3) Centroid dari setiap cluster akan diperbarui dengan menjumlahkan semua koordinat objek dan membaginya dengan jumlah objek dalam cluster.

$$C_j = \frac{1}{|S_j|} \sum_{X_i \in S_j} X_i \quad (2)$$

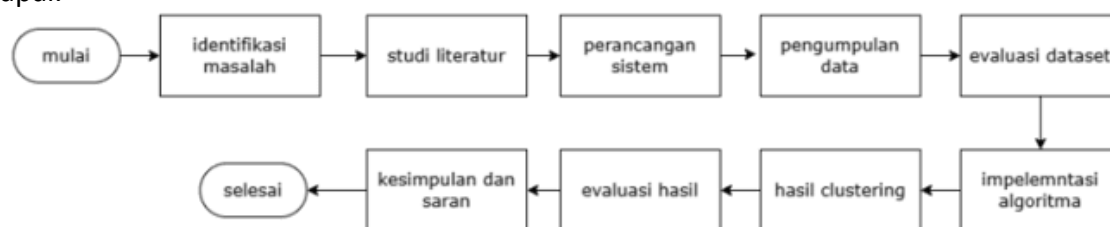
Di mana S_j adalah kumpulan objek dalam cluster j , pembaruan centroid ini bertujuan untuk menggerakkan centroid ke posisi yang lebih mirip dengan seluruh objek dalam cluster, sehingga setiap iterasi mendekatkan solusi ke konfigurasi cluster yang ideal.

- 4) Algoritma K-Means mengulang proses pengelompokan dan memperbarui centroid hingga konvergensi tercapai. Jumlah jarak objek dihitung ulang setiap iterasi. Jika centroid yang lebih dekat ditemukan, objek dapat berpindah ke cluster. Proses berulang hingga posisi centroid stabil atau jumlah iterasi maksimum dicapai. Ini memastikan bahwa cluster yang dihasilkan memenuhi kriteria jarak minimum.
- 5) Ketika centroid tidak berubah lagi atau tidak ada objek yang berpindah cluster antar iterasi, K-Means dianggap konvergen. Pada saat ini, cluster telah stabil dan objeknya lebih mirip satu sama lain daripada objek di cluster lain, yang menunjukkan bahwa algoritma berhasil mencapai tujuannya.

3. METODE PENELITIAN

3.1 Kerangka Alur Penelitian

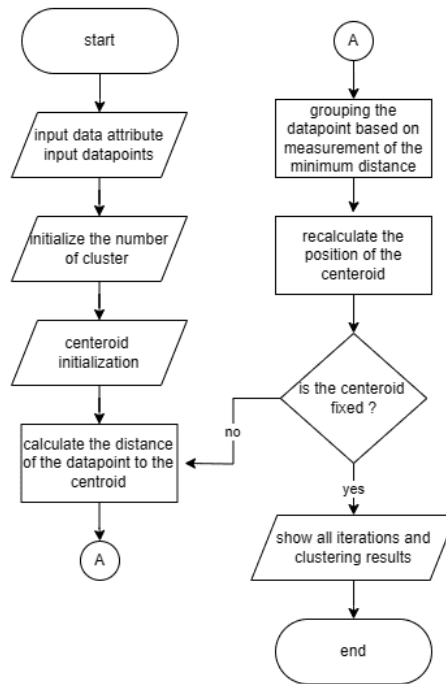
Penelitian ini dimulai dengan menemukan masalah utama yang akan difokuskan; ini termasuk memilih dan mengelompokkan *framework* dan *library* PHP berdasarkan atribut kode. Selain itu, penelitian literatur dilakukan untuk menemukan referensi tentang penerapan algoritma K-Means dalam kelompok dan struktur yang relevan. Untuk mendukung penelitian, desain algoritma, struktur data, dan alat disusun selama tahap perancangan sistem. Proses pengumpulan data dimulai dengan menyiapkan dataset yang dibutuhkan untuk clustering. Kemudian, kualitas data diperiksa untuk menemukan atribut yang relevan. Pada tahap implementasi, algoritma K-Means digunakan untuk mengelompokkan data berdasarkan karakteristik yang sudah ada. Hasil clustering kemudian dianalisis, menghasilkan kluster-kluster menggunakan algoritma K-Means dan menemukan kluster mana yang terbentuk. Selanjutnya, hasil clustering dievaluasi untuk menilai validitas dan kualitasnya. Hasil evaluasi, kesimpulan, dan saran terkait penggunaan *framework* atau *library* berdasarkan hasil *clustering*. Setelah itu, proses penelitian berakhir, menunjukkan bahwa tujuannya telah dicapai.



Gambar 3. Flowchart Kerangka

3.2 Flowchart Perhitungan K-Means

Flowchart perhitungan yang digunakan dalam sistem dengan menerapkan algoritma K-Means dapat dilihat pada gambar 4. Sistem akan dimulai dengan mendaftarkan datapoint dan fitur yang akan dianalisis. Selanjutnya, pengguna menentukan jumlah cluster yang diinginkan dan kemudian memulai centroid. Setiap datapoint kemudian dihitung jarak terhadap setiap centroid untuk menentukan cluster awal berdasarkan jarak terdekat. Kemudian, posisi centroid dihitung ulang dan langkah-langkah ini diulangi hingga tidak ada perubahan posisi. Setelah kondisi ini tercapai, output akan menampilkan seluruh iterasi dan hasil clustering akhir.



Gambar 4. Flowchart Algoritma K-Means Pada Sistem

3.3 Attribute

Dalam implementasi algoritma K-Means memerlukan *attribute* sebagai variabel yang akan menjadi salah satu faktor analisa *code structure*, dalam penelitian ini digunakan 13 *attribute* sebagai parameter yang akan digunakan dalam perhitungan K-Means, dapat dilihat dari tabel 1. Yang menggambarkan *attribute* yang akan digunakan.

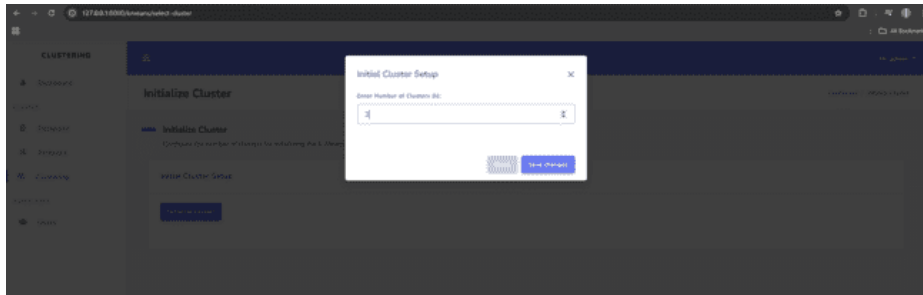
Tabel 1. Data *Attribute*

Kode <i>Attribute</i>	Nama <i>Attribute</i>	Deskripsi <i>Attribute</i>
A1	<i>Directories</i>	Jumlah Folder
A2	<i>Files</i>	Jumlah File
A3	<i>Lines of Code (LOC)</i>	Jumlah baris kode
A4	<i>Comment Lines of Code (CLOC)</i>	Jumlah baris kode sebagai komentar
A5	<i>Non-Comment Lines of Code (NCLOC)</i>	Jumlah baris kode yang bukan komentar
A6	<i>Logical Lines of Code (LLOC)</i>	Jumlah baris kode logika
A7	<i>Size-Classes</i>	Jumlah kelas
A8	<i>Average Class Length</i>	Rata-rata panjang baris kode kelas
A9	<i>Average Method Length</i>	Rata-rata jumlah baris kode pada fungsi/metode
A10	<i>Average Methods Per Class</i>	Rata-rata Jumlah baris kode fungsi/metode
A11	<i>Functions</i>	Jumlah Fungsi
A12	<i>Average Function Length</i>	Rata-rata Jumlah baris kode pada fungsi
A13	<i>Not in classes or functions</i>	<i>Jumlah baris kode tidak di dalam kelas atau fungsi</i>

4. HASIL DAN PEMBAHASAN

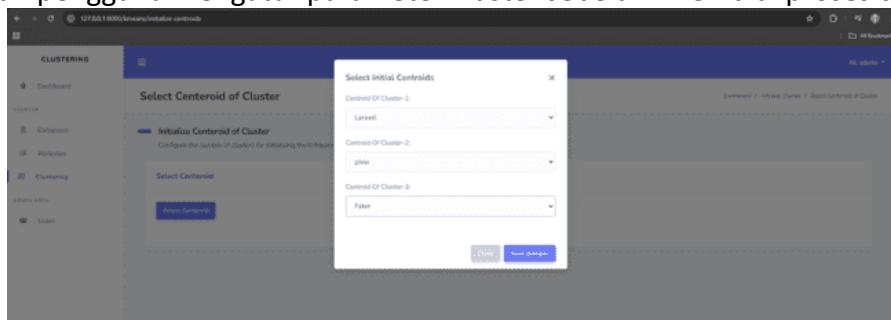
4.1. Hasil Implementasi

Aplikasi yang menerapkan algoritma k-means berbasis website berhasil diterapkan dalam penelitian ini dengan menunjukkan beberapa antarmuka yang



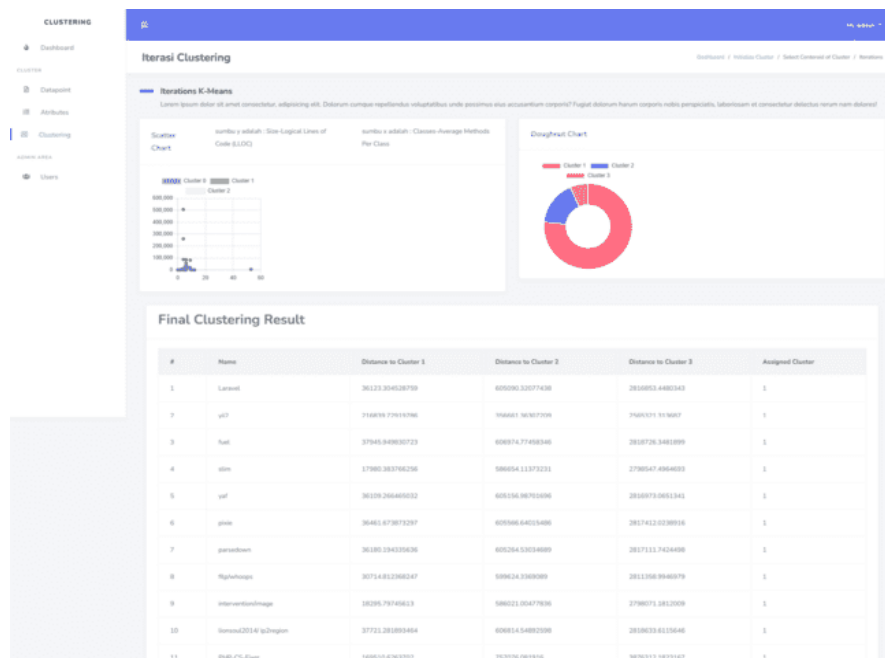
Gambar 5. Menentukan Jumlah Cluster (Sistem)

Gambar 5 menunjukkan modal dialog yang memungkinkan pengguna untuk memasukkan jumlah kluster yang diinginkan melalui kolom input, dengan dua tombol aksi: Close untuk membatalkan dan Save Changes untuk menyimpan pengaturan. Antarmuka ini memudahkan pengguna mengatur parameter kluster sebelum memulai proses clustering.



Gambar 6. Inisialisasi Centroid (Sistem)

Gambar 6 menunjukkan modal dialog yang memungkinkan pengguna memilih centroid awal untuk setiap kluster melalui dropdown menu, dengan dua tombol aksi: Close untuk membatalkan dan Save Changes untuk menyimpan pilihan. Antarmuka ini memudahkan pengguna mengatur parameter awal kluster sebelum memulai proses clustering.



Gambar 7. Proses Dan Hasil Clustering (Sistem)

Gambar 7 menampilkan visualisasi hasil clustering dalam bentuk doughnut dan scatter chart untuk mempermudah pemahaman. Tersedia pula tabel final cluster hasil konvergensi dari iterasi K-Means, serta tabel centroid dan jarak pada setiap iterasi perhitungan K-Means.

4.2. Pengumpulan Data

Pada penelitian ini dilakukan proses pengumpulan data menggunakan metode *scraping* menggunakan *tools* "phpLoc", untuk dapat menghasilkan data yang dibutuhkan. Jumlah data yang terkumpul adalah 158 *datapoint* dengan 53 baris data berkategori sebagai "*framework*" dan 105 baris data berkategori sebagai "*Library*". Untuk melakukan uji coba algoritma K-Means peneliti menggunakan data sample 30 *row* dari dataset yang sudah dikumpulkan, data sample dapat dilihat pada tabel 2. Sebagai berikut:

Tabel 2. Data sample Algoritma K-Means

Data Point	A1	A2	A3	A4	...	A12	A13
Laravel	18	55	2889	1155	...	2	35
yii	315	1998	820072	60416	...	4	4405
yii2	176	1018	197727	59452	...	0	237
fuel	10	16	1407	808	...	20	33
cakephp	339	1535	357242	105771	...	0	551
...
sage	10	27	681	190	...	3	12

4.3. Hasil Perhitungan Manual

Dalam penelitian ini, perhitungan manual dilakukan menggunakan data sampel yang terdiri dari 30 baris data yang diambil dari total 158 *datapoint*. Dataset ini mencakup 7 *framework* dan 23 *library*. Clustering dilakukan dengan menentukan tiga cluster, dengan centroid awal masing-masing cluster adalah "Laravel", "pixie", dan "joomla-cms". Proses ini dilakukan berulang kali hingga mencapai kondisi konvergensi, dengan centroid pada iterasi terakhir adalah hasil akhir dari proses. Tabel 2. menggambarkan dari iterasi sebelumnya serta tabel 3. menggambarkan antara setiap objek dengan centroid masing-masing cluster akan disajikan sebagai bagian dari dokumentasi perhitungan manual. Tujuan proses ini adalah untuk mengelompokkan data sesuai dengan fitur yang telah ditentukan, dan untuk membandingkannya dengan hasil clustering yang dihasilkan oleh sistem yang digunakan.

Tabel 3. Hasil Manual Perhitungan Jarak Dari Iterasi Terakhir

No.	Nama	Distance to Cluster 1	Distance to Cluster 2	Distance to Cluster 3	Assigned Cluster
1	Laravel	36123.30	605090.32	2816853.45	1
2	Yii2	216839.73	356661.36	2565321.31	1
3	fuel	37945.95	606974.77	2818726.35	1
4	slim	17980.38	586654.11	2798547.50	1
...
30	magento2	3604299.21	3038331.43	822765.03	3

Tabel 4. Hasil Manual Perhitungan Centeroid Baru Dari Iterasi Terakhir

Attribute	Centeroid 1	Centeroid 2	Centeroid 3
A1	39.83	522.60	5572.00
A2	208.30	2109.40	16757.00
A3	25328.17	464600.00	2200091.00
A4	8225.70	95538.60	647894.00
A5	27641.52	369061.40	1552197.00
A6	6450.65	70907.80	383518.50
A7	6376.65	63779.60	370594.50
A8	53.17	43.40	24.50
A9	3.78	5.80	5.00
A10	9.78	6.20	4.00
A11	16.91	621.60	679.00
A12	1.48	1.80	0.00
A13	57	6506.60	12245.00

4.4. Hasil Perhitungan Menggunakan Sistem

Dataset yang sama digunakan untuk perhitungan dengan sistem. Algoritma K-Means yang ada dalam sistem digunakan untuk melakukan clustering, yang menggunakan tiga cluster dengan centroid awal yang sama seperti perhitungan manual. Tabel 4. Akan menggambarkan jarak objek ke *centroid* dan tabel 5. Akan menggambarkan *centroid* akhir menyajikan hasil sistem, yang digunakan untuk membandingkan akurasi dan konsistensi hasil manual dan memvalidasi implementasi algoritma dalam sistem.

Tabel 5. Hasil Sistem Perhitungan Jarak Dari Iterasi Terakhir

No.	Nama	Distance to Cluster 1	Distance to Cluster 2	Distance to Cluster 3	Assigned Cluster
1	Laravel	36123.304528759	605090.32077438	2816853.4480343	1
2	yii2	216839.72919786	356661.36307209	2565321.313687	1
3	fuel	37945.949830723	606974.77458346	2818726.3481899	1
4	slim	17980.383766256	586654.11373231	2798547.4964693	1
...
30	magento2	3604299.205012	3038331.4288617	822765.0314736	3

Tabel 6. Hasil Sistem Perhitungan Centeroid Baru Dari Iterasi Terakhir

Attribute	Centeroid 1	Centeroid 2	Centeroid 3
A1	39.826086956522	522.6	5572
A2	208.30434782609	2109.4	16757
A3	25328.173913043	464600	2200091
A4	8225.6956521739	95538.6	647894
A5	27641.52173913	369061.4	1552197
A6	6450.652173913	70907.8	383518
A7	6376.652173913	63779.6	370594
A8	53.173913043478	43.4	24
A9	3.7826086956522	5.8	5
A10	9.7826086956522	6.2	4

A11	16.913043478261	621.6	679
A12	1.4782608695652	1.8	0
A13	57	6506.6	12245

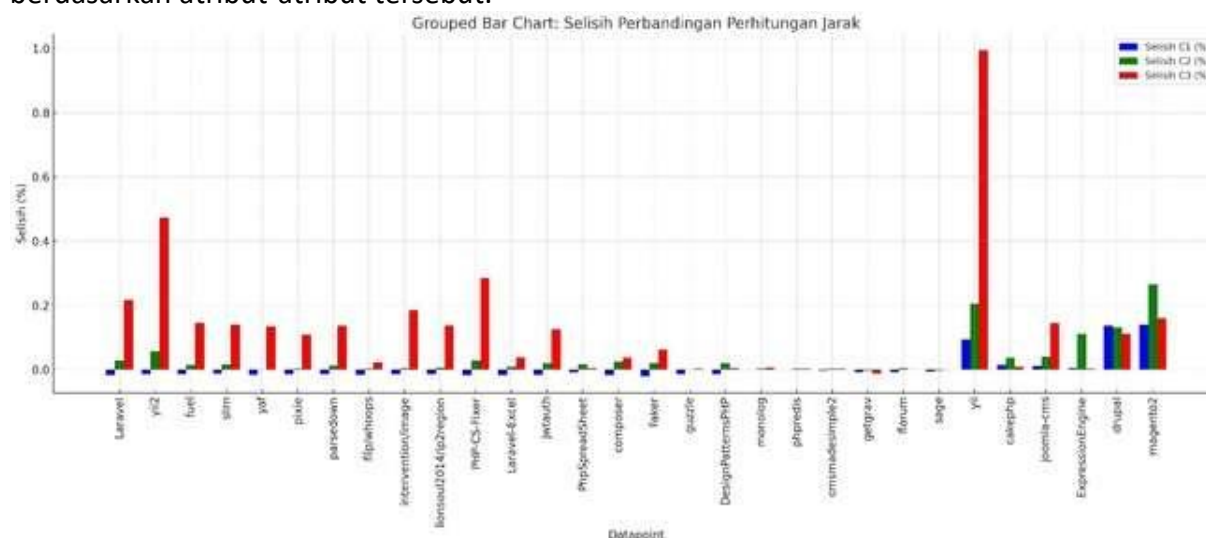
4.5. Evaluasi Hasil

Pada iterasi terakhir, hasil klasterisasi manual dan sistem menunjukkan konsistensi, terutama pada datapoint seperti Laravel, Yii2, Slim, dan ParseDown yang berada di Klaster 1 dengan selisih kecil (-0.01% hingga -0.02% untuk Selisih C1). Namun, terdapat perbedaan signifikan pada beberapa datapoint, seperti Yii (Selisih C3 tertinggi 0.994%), Drupal, dan Magento2, yang secara konsisten dikelompokkan ke Klaster 2 dan Klaster 3 berdasarkan nilai selisih dominan.

Tabel 7. Selisih Perbandingan Perhitungan Jarak

Datapoint	Selisih C1 (%)	Selisih C2 (%)	Selisih C3 (%)	Result (Manual)	Result (Sistem)
Laravel	-0.017%	0.028%	0.216%	Klaster 1	Klaster 1
yii2	-0.014%	0.057%	0.473%	Klaster 1	Klaster 1
fuel	-0.015%	0.015%	0.145%	Klaster 1	Klaster 1
slim	-0.012%	0.016%	0.139%	Klaster 1	Klaster 1
...
cakephp	0.014%	0.036%	0.009%	Klaster 2	Klaster 2
joomla-cms	0.011%	0.041%	0.144%	Klaster 2	Klaster 2
ExpressionEngine	0.004%	0.110%	0.003%	Klaster 2	Klaster 2
drupal	0.137%	0.132%	0.110%	Klaster 3	Klaster 3
magento2	0.140%	0.265%	0.160%	Klaster 3	Klaster 3

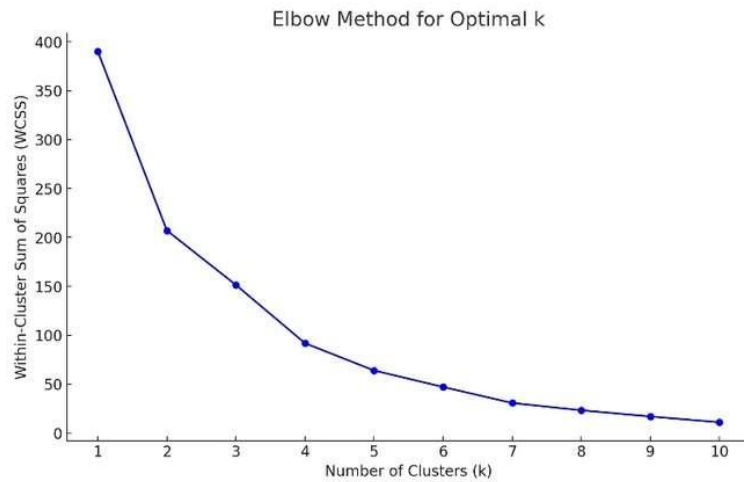
Pada gambar 8. menunjukkan selisih yang ditampilkan dalam persen ini memberikan gambaran tentang seberapa besar perbedaan antara hasil clustering manual dan hasil sistem berdasarkan atribut-atribut tersebut.



Gambar 8. Selisih Perbandingan Perhitungan Jarak

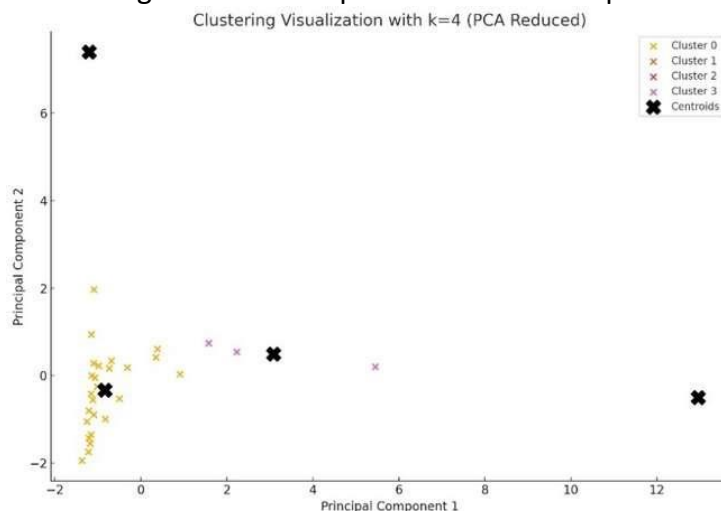
Visualisasi Gambar 8 menunjukkan bahwa dominasi selisih pada salah satu klaster menjadi indikator utama penentuan klaster datapoint. Misalnya, Yii2 menonjol pada Selisih C3, sementara ExpressionEngine dan CakePHP memiliki selisih yang lebih merata namun tetap mengarah ke Klaster 2. Hasil ini menegaskan keakuratan sistem dalam mereplikasi metode

manual dengan kesalahan minimal, menunjukkan efektivitas algoritma dalam pengelompokan datapoint sesuai hasil manual.



Gambar 9. Hasil Perhitungan Elbow Method

Gambar 9 menunjukkan grafik Elbow Method untuk menentukan jumlah cluster optimal dalam proses clustering. Sumbu x merepresentasikan jumlah cluster (k), sedangkan sumbu y menunjukkan nilai *Within-Cluster Sum of Squares* (WCSS). Penurunan tajam WCSS terlihat di awal grafik, tetapi setelah $k=4$, penurunan menjadi kecil, membentuk pola *elbow*. Pada $k=4$, nilai WCSS adalah 91.799, dengan penurunan signifikan sebesar 59.577 dari $k=3$ (151.376). Namun, dari $k=4$ ke $k=5$ (63.954), penurunan hanya 27.845, menunjukkan pengurangan varians semakin kecil. Oleh karena itu, $k=4$ dipilih sebagai jumlah cluster optimal karena mencapai keseimbangan antara kompleksitas model dan performa clustering.



Gambar 10. Hasil Clustering Berdasarkan Elbow Method

Gambar 10 menunjukkan visualisasi hasil clustering dengan $k=4$, di mana data telah direduksi menjadi dua dimensi menggunakan metode *Principal Component Analysis* (PCA) untuk mempermudah interpretasi. Setiap warna merepresentasikan satu cluster, dengan titik-titik yang menunjukkan data individu dalam dataset. Centroid dari masing-masing cluster ditampilkan sebagai simbol "X" berwarna hitam, yang merepresentasikan pusat geometris cluster tersebut. Grafik ini mengilustrasikan pemisahan yang cukup baik antara cluster,

menunjukkan bahwa algoritma K-Means mampu mengelompokkan data dengan karakteristik yang serupa ke dalam cluster yang sama. Cluster yang lebih padat dan terpisah menandakan tingkat kohesi dan separasi yang baik, sesuai dengan hasil evaluasi Silhouette Score sebelumnya. Visualisasi ini memberikan gambaran intuitif tentang distribusi data dalam masing-masing cluster serta hubungan antar cluster.

5. KESIMPULAN

Penelitian ini bertujuan untuk mengelompokkan framework dan library PHP berdasarkan analisis struktur kode menggunakan algoritma K-Means dengan sistem yang dibuat berbasis website. Pendekatan yang digunakan melibatkan analisis 13 atribut, seperti LOC, NCLOC, CLOC, rata-rata panjang kelas, panjang metode, dan distribusi fungsi, untuk menemukan pola distribusi dalam kelompok yang signifikan. Hasil penelitian menunjukkan bahwa dalam pengujian metode, baik dengan perhitungan manual maupun sistem berbasis website, jumlah cluster yang optimal adalah $k=3$ meskipun *Elbow Method* menunjukkan bahwa $k=4$ adalah jumlah cluster optimal berdasarkan penurunan nilai WCSS yang signifikan.

Meskipun *Elbow Method* menunjukkan $k=4$ sebagai jumlah cluster optimal, analisis lebih lanjut dengan menggunakan *Silhouette Score* dan evaluasi visual menunjukkan bahwa pembagian data menjadi 3 cluster memberikan pemisahan yang lebih jelas dan lebih mudah diinterpretasikan. Cluster 1 mewakili framework dengan struktur kode yang kecil dan sederhana, sementara Cluster 2 dan Cluster 3 mencerminkan framework dengan struktur yang lebih kompleks dan ukuran kode yang lebih besar. Oleh karena itu, dalam penelitian ini, jumlah cluster 3 dipilih sebagai hasil yang paling optimal, karena memberikan keseimbangan yang lebih baik antara kompleksitas model dan interpretasi yang jelas.

Selain itu, hasil clustering menggunakan sistem berbasis website menunjukkan presisi lebih tinggi, dengan perhitungan jarak yang mencapai hingga 15 angka di belakang koma, dibandingkan dengan perhitungan manual yang hanya mencapai 5-6 angka, yang mengindikasikan peningkatan akurasi. Penelitian ini memberikan kontribusi pada pengelompokan framework PHP yang lebih efisien dan mendukung pengembang dalam memilih tools sesuai kebutuhan. Namun, penelitian ini masih memiliki keterbatasan dalam pemilihan centroid yang dilakukan secara acak. Oleh karena itu, disarankan untuk menggunakan algoritma K-Means++ untuk pemilihan centroid yang lebih konsisten, menyediakan scatter chart dinamis untuk visualisasi yang lebih fleksibel dari sistem yang dibuat, serta mengintegrasikan *Silhouette Score* untuk evaluasi jumlah cluster yang optimal ketika pengguna memasukkan dataset mereka. Dengan demikian, penelitian ini tidak hanya berkontribusi secara teknis tetapi juga memberikan wawasan baru bagi pengembangan perangkat lunak berbasis PHP.

REFERENSI

- [1] Yudhistira, Y., & Wardoyo, R. (2023). *Choosing the right technology stack for modern applications*. International Journal of Software Engineering and Applications.
- [2] Benmoussa, A., Bouhnik, D., & Messaoudi, S. (2020). Frameworks for web development: A comparative study. Journal of Software Engineering and Applications.
- [3] Kansha, W. M., Rahman, A., & Yudhistira, Y. (2023). Impact of framework selection on software development efficiency. Journal of Software Development and Technology.

- [4] Endra, R. Y., Rahman, A., & Sudrajat, A. (2021). Code structure analysis in web frameworks: A case study on Laravel. *International Journal of Computer Science and Information Security*.
- [5] Benmoussa, A., Bouhnik, D., & Messaoudi, S. (2019). Frameworks for web development: A comparative study. *Journal of Software Engineering and Applications*.
- [6] Kansha, W. M., Saherih, & Muchlis. (2023). Analisis perbandingan struktur dan performa framework CodeIgniter dan Laravel dalam pengembangan web application. *Jurnal Teknik Informatika STMIK Antar Bangsa*, 9(1), 25-28
- [7] Wicaksono, H. (2020). Library 2.0 dan dampaknya dalam pengembangan aplikasi dan layanan perpustakaan. *BACA*, 31(1), 15-26.
- [8] Wicaksono, H. (2020). Library 2.0 dan dampaknya dalam pengembangan aplikasi dan layanan perpustakaan. *BACA*, 31(1), 15-26.
- [9] Nugraha, A., Nurdiawan, O., & Dwilestari, G. (2022). Penerapan Data Mining Metode K-Means Clustering Untuk Analisa Penjualan Pada Toko Yana Sport. *JATI (Jurnal Mahasiswa Teknik Informatika*, 6(2), 849–855.