

Penerapan K-means Clustering untuk Pengukuran Kinerja Programmer di Software House

Alya Fitria¹, Taufik Ardiansyah Putra², Syahiduz Zaman³

^{1,2,3}Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang

*Email: ¹200605110022@student.uin-malang.ac.id, ²200605110071@student.uin-malang.ac.id, ³syahid@ti.uin-malang.ac.id

Abstrak

Evaluasi kinerja para programmer memiliki peran penting dalam menjaga efisiensi sebuah perusahaan pengembang perangkat lunak. Penelitian ini mengusulkan penerapan K-means clustering sebagai metode untuk mengukur dan mengkategorikan kinerja programmer berdasarkan beberapa kriteria. Pendekatan yang diajukan melibatkan penilaian terhadap kualitas kode, produktivitas, kemampuan teknis, kolaborasi tim, dan kemampuan dalam pemecahan masalah. Dengan menerapkan metode K-means clustering, para programmer dapat dikelompokkan ke dalam kluster kinerja yang berbeda, memungkinkan identifikasi antara programmer dengan performa tinggi, sedang, dan dalam tahap perkembangan. Metode K-means clustering membagi data menjadi kelompok yang berkaitan dengan menghitung jarak antara titik data dan pusat kelompok, dan diulang hingga kelompok yang stabil terbentuk.

Kata kunci: Clustering, K-Means, Kinerja, Programmer, SDM

Abstract

Performance evaluation of programmers plays a crucial role in maintaining efficiency within a software development company. This study proposes the implementation of K-means clustering as a method to measure and categorize programmer performance based on several criteria. The proposed approach involves assessing code quality, productivity, technical skills, team collaboration, and problem-solving abilities. By applying the K-means clustering method, programmers can be grouped into different performance clusters, allowing for the identification of high, moderate, and developmental performers. The K-means clustering method divides data into clusters related by calculating the distance between data points and cluster centers, and iterates until stable clusters are formed..

Kata kunci: Clustering, K-Means, Kinerja, Programmer, SDM

PENDAHULUAN

Dalam era digital dan teknologi informasi yang terus berkembang pesat, perusahaan-perusahaan di bidang teknologi, khususnya perusahaan pengembang perangkat lunak atau yang dikenal sebagai "software house," memiliki peran sentral dalam menyediakan solusi perangkat lunak yang inovatif dan berkualitas tinggi. Software house adalah entitas yang berfokus pada pengembangan aplikasi perangkat lunak, sistem informasi, dan solusi teknologi untuk memenuhi kebutuhan pelanggan mereka.

Manajemen sumber daya manusia (SDM) adalah salah satu aspek kunci dalam operasi software house. SDM yang berkualitas adalah aset berharga bagi perusahaan, terutama dalam industri perangkat lunak, dimana keahlian

teknis dan kreativitas menjadi faktor kunci dalam menghasilkan solusi yang unggul. Oleh karena itu, penting bagi perusahaan software house untuk mengelola dan mengevaluasi kinerja para programmer dengan cermat.

Evaluasi kinerja programmer adalah salah satu tugas penting dalam manajemen SDM di software house. Evaluasi ini bukan hanya tentang menentukan seberapa baik atau buruk seorang programmer dalam menghasilkan kode, tetapi juga tentang memahami bagaimana mereka berkontribusi pada proyek, berkolaborasi dalam tim, dan menyelesaikan masalah teknis yang mungkin muncul.

Source code adalah bagian yang esensial dalam proyek perangkat lunak (Saputra *et al.*, 2021). Dalam penelitian ini, pengukuran kualitas kode diterjemahkan ke dalam empat skala

penilaian yang mempengaruhi stabilitas, keandalan, dan kinerja perangkat lunak yang dihasilkan. Skala pertama adalah "Sangat Baik," yang mencirikan kode yang sangat bersih, efisien, dan mudah dipahami. Skala kedua adalah "Baik," menunjukkan kode yang cukup baik, meskipun mungkin memerlukan beberapa perbaikan kecil. Skala ketiga adalah "Cukup," menggambarkan kode yang membutuhkan perbaikan signifikan untuk memastikan kualitas yang memadai. Skala terakhir adalah "Kurang Baik," yang mengindikasikan kode yang memerlukan perbaikan segera karena berpotensi merugikan stabilitas dan kinerja perangkat lunak. Dengan demikian, dalam evaluasi kinerja programmer, aspek kualitas kode memiliki peran kunci dalam mencapai keberhasilan proyek perangkat lunak.

Produktivitas kerja karyawan memainkan peran krusial dalam mendukung kesuksesan usaha (Wahyuningsih, 2019). Penelitian ini mengukur produktivitas dengan menggunakan empat skala penilaian yang berpengaruh pada efisiensi penggunaan waktu, sumber daya, dan kemampuan menghasilkan output dalam waktu tertentu. Skala pertama adalah "Sangat Tinggi," menggambarkan tingkat produktivitas yang luar biasa, yang mampu mengurangi biaya operasional secara signifikan, meningkatkan daya saing perusahaan, dan memberikan keunggulan kompetitif di pasar. Skala kedua adalah "Tinggi," mencerminkan produktivitas yang tinggi, meskipun masih ada ruang untuk peningkatan. Skala ketiga adalah "Cukup," menunjukkan tingkat produktivitas yang memerlukan peningkatan yang cukup besar untuk mencapai efisiensi yang diinginkan. Skala terakhir adalah "Rendah," mengindikasikan produktivitas yang rendah dan memerlukan tindakan perbaikan segera untuk menghindari dampak negatif pada kemajuan proyek dan pencapaian tujuan perusahaan. Dalam konteks evaluasi kinerja programmer, produktivitas mereka dalam menghasilkan kode atau solusi perangkat lunak menjadi indikator penting karena berdampak pada kemajuan proyek dan pencapaian tujuan perusahaan.

Hard skill merupakan penguasaan ilmu pengetahuan, teknologi dan keterampilan teknis yang berhubungan dengan bidang ilmunya (Syahputra, 2021). Sehingga dalam hal

pengukuran kemampuan teknis atau hard skill pada programmer digunakan empat skala penilaian yang mencerminkan tingkat penguasaan dalam berbagai aspek teknis. Skala pertama adalah "Sangat Tinggi," yang menggambarkan kemampuan teknis yang sangat mendalam dan komprehensif, termasuk pengetahuan yang mendalam tentang bahasa pemrograman, arsitektur perangkat lunak, alat pengembangan, dan teknologi terbaru yang mendukung produksi solusi perangkat lunak berkualitas tinggi. Skala kedua adalah "Tinggi," yang menunjukkan tingkat kemampuan teknis yang baik, meskipun masih ada beberapa area yang dapat ditingkatkan. Skala ketiga adalah "Cukup," mengindikasikan kemampuan teknis yang memerlukan peningkatan yang signifikan untuk mencapai standar yang diharapkan. Skala terakhir adalah "Rendah," yang mencerminkan tingkat kemampuan teknis yang jauh dari yang diharapkan dan memerlukan tindakan perbaikan segera untuk mendukung kesuksesan proyek perangkat lunak. Dalam konteks evaluasi kinerja programmer, kemampuan teknis yang kuat menjadi faktor kunci dalam menentukan kemampuan seorang programmer untuk mengatasi tugas-tugas teknis dan berkontribusi pada kesuksesan proyek perangkat lunak.

Kolaborasi tim, yang dipaparkan dalam kutipan dari penelitian (Sutisna, 2019), memiliki dampak yang signifikan pada kemajuan perusahaan. Dalam penelitian ini, pengukuran kolaborasi tim dilakukan menggunakan empat skala penilaian yang mencerminkan berbagai tingkat kolaborasi dalam tim. Skala pertama adalah "Sangat Tinggi," yang menunjukkan kolaborasi tim yang sangat efektif dimana anggota tim bekerja secara sinergis, berbagi sumber daya secara maksimal, dan mencapai tujuan bersama secara konsisten. Skala kedua adalah "Tinggi," mengindikasikan kolaborasi yang baik, meskipun ada sedikit ruang untuk perbaikan. Skala ketiga adalah "Cukup," yang menandakan adanya peluang signifikan untuk meningkatkan kolaborasi tim, dengan beberapa tantangan yang perlu diatasi. Skala terakhir adalah "Rendah," yang mencerminkan kolaborasi yang sangat rendah dan memerlukan tindakan segera untuk memperbaiki hubungan dan meningkatkan efektivitas tim. Dalam konteks evaluasi kinerja programmer, kolaborasi tim menjadi faktor penting karena tim yang

dapat berkolaborasi dengan baik cenderung lebih efektif dalam menyelesaikan tugas, memecahkan masalah, dan menghasilkan produk perangkat lunak yang lebih baik.

Problem solving, sebagaimana dikutip dalam (Toharudin and Setia Kurniawan, 2017), merupakan inti dari proses berpikir yang memerlukan kemampuan untuk memecahkan masalah. Dalam penelitian ini, evaluasi kinerja programmer melibatkan pengukuran kemampuan pemecahan masalah dalam empat skala penilaian yang berbeda. Skala pertama adalah "Pemecahan Masalah Tingkat Tinggi," yang mencirikan programmer yang secara konsisten mampu mengidentifikasi masalah dengan cepat, menganalisisnya secara mendalam, dan menemukan solusi yang inovatif dan efektif. Skala kedua adalah "Pemecahan Masalah Tingkat Menengah," yang menggambarkan programmer yang memiliki kemampuan pemecahan masalah yang baik, meskipun ada ruang untuk peningkatan dalam kreativitas solusi. Skala ketiga adalah "Pemecahan Masalah Dasar," mengindikasikan bahwa programmer memiliki kemampuan pemecahan masalah yang terbatas dan memerlukan bimbingan lebih lanjut. Skala terakhir adalah "Pemecahan Masalah Rendah," yang mencerminkan kemampuan pemecahan masalah yang sangat terbatas atau bahkan tidak ada. Dalam evaluasi kinerja programmer, kemampuan pemecahan masalah menjadi indikator penting karena programmer yang memiliki kemampuan ini cenderung lebih efektif dalam mengatasi tantangan teknis, menemukan solusi yang kreatif, dan menjaga kelancaran proyek perangkat lunak.

Evaluasi kinerja yang baik dapat membantu perusahaan dalam menjaga kualitas produk yang dihasilkan. Tingkat kualitas produk perangkat lunak oleh sebuah software house memiliki dampak yang sangat signifikan terhadap citra perusahaan, kepuasan pelanggan, dan kelangsungan bisnis. Kualitas kode yang diproduksi oleh para programmer menjadi fondasi dari produk perangkat lunak tersebut. Dengan kata lain, kode yang baik akan menghasilkan produk yang andal, aman, dan sesuai dengan harapan pengguna. Oleh karena itu, menjaga kualitas produk adalah salah satu fokus utama bagi software house.

Untuk mencapai kualitas produk yang optimal, tidak hanya kompetensi teknis yang diperlukan, tetapi juga manajemen sumber daya manusia yang efisien. Ini berkaitan dengan pengoptimalan penggunaan sumber daya internal seperti programmer yang terlibat dalam pengembangan perangkat lunak. Dengan mengidentifikasi dan mengelola dengan baik programmer yang berkinerja tinggi, perusahaan dapat mengalokasikan sumber daya dengan lebih tepat, memberikan tanggung jawab yang sesuai, dan memastikan bahwa tim bekerja secara efisien dalam mencapai tujuan pengembangan. Namun, mengevaluasi kinerja programmer dengan cara yang objektif dan konsisten bisa menjadi tantangan. Hal ini terutama berlaku jika perusahaan memiliki banyak programmer dan berbagai kriteria penilaian yang kompleks.

Telah banyak penelitian sebelumnya yang berhubungan dengan klusterisasi menggunakan algoritma K-Means. Sebagai contoh, dalam penelitian berjudul "Klusterisasi Pendidikan Masyarakat untuk Mengidentifikasi Daerah dengan Pendidikan Terendah Menggunakan Algoritma K-Means" (Nurahman and Dwi Aulia, 2023), dilakukan analisis menggunakan 230 sampel data dengan 11 parameter, yang kemudian direduksi menjadi 229 sampel data dengan 10 parameter, berdasarkan tingkat pendidikan. Tingkat pendidikan mencakup Tidak/belum sekolah, Belum tamat SD, Tamat SD, SLTP, SLTA, D1 dan D2, D3, S1, S2, dan S3. Hasil penelitian ini menghasilkan delapan kluster dengan distribusi data yang berbeda pada masing-masing kluster.

Penelitian lain yang menggunakan metode K-Means Clustering terdapat dalam judul "Pemetaan Kelompok Kinerja Siswa Menggunakan Metode Clustering K-Means dan Algoritma Greedy" (Amirulloh, 2019). Penelitian ini melibatkan 37 sampel data siswa dengan parameter berupa nilai-nilai siswa sebelumnya, dengan fokus pada variabel sikap, pengetahuan, dan keterampilan dari setiap mata pelajaran. Hasil analisis menunjukkan adanya lima kluster dengan distribusi jumlah siswa dalam masing-masing kluster.

Penelitian terkait lain yang berjudul "Pengelompokan Kualitas Kinerja Pegawai Menggunakan Metode K-Means Clustering" (Thaher, Septiarini and Puspitasari, 2022) menggunakan 25 sampel data pegawai dengan

parameter penilaian profesional, integritas, amanah, capaian kinerja pegawai, dan absensi. Hasil penelitian ini menghasilkan tiga klaster dengan distribusi karyawan dalam setiap klaster.

Penelitian ini, dengan judul "Penerapan K-Means Clustering untuk Pengukuran Kinerja Programmer di Software House" akan mengambil pendekatan yang berfokus pada lingkungan software house, dengan kriteria penilaian yang berbeda. Penelitian ini memberikan kelanjutan yang lebih baik daripada penelitian sebelumnya dengan tujuan yang serupa, seperti dalam penelitian berjudul "Sistem Pendukung Keputusan Penilaian Kinerja Karyawan Menggunakan Metode TOPSIS di PT. Karya Mitra Nugraha" (Mauludin and Suali, 2019). Adapun pembaruan atau perbedaan lain dari penelitian ini yaitu menggunakan metode Clustering dalam mengevaluasi kinerja karyawan dengan kriteria penilaian yang berbeda.

Dengan demikian, penelitian ini akan memberikan kontribusi yang lebih mendalam dalam memahami evaluasi kinerja programmer dalam konteks yang lebih spesifik dan relevan dalam lingkungan software house.

METODE PENELITIAN

Tahapan Penelitian

Pada pelaksanaannya, penelitian ini melakukan beberapa tahapan proses, yang terdiri dari identifikasi masalah, menentukan tujuan, pengumpulan data, penerapan k-means dan kesimpulan.



Gambar 1. Tahapan Penelitian

1. Identifikasi Masalah: Tahap pertama penulis mengidentifikasi masalah kinerja programmer di software house.
2. Menentukan Tujuan: Tujuan dari penelitian ini adalah untuk mengukur kinerja programmer menggunakan metode K-Means Clustering.
3. Pengumpulan Data: Tahap ini melibatkan pengumpulan data kinerja programmer di software house berdasarkan parameter-parameter yang telah ditentukan.
4. Penerapan K-Means Clustering: Tahap utama dalam penelitian ini adalah penerapan algoritma K-Means Clustering untuk mengelompokkan programmer ke dalam kelompok berdasarkan parameter-parameter yang sudah ditentukan.
5. Kesimpulan: Pada tahapan penelitian ini penulis menentukan kesimpulan dari hasil yang dicapai serta memberikan rekomendasi atau saran untuk perbaikan di masa depan.

K-Means

K-Means Clustering adalah salah satu metode clustering non-hirarki yang digunakan dalam analisis data untuk mengelompokkan data menjadi beberapa kelompok berdasarkan kemiripan karakteristik. Metode ini sangat berguna dalam memahami struktur data dan mengidentifikasi pola yang ada. K-Means berusaha untuk mengelompokkan data sedemikian rupa sehingga data dalam satu kelompok memiliki kesamaan yang tinggi, sementara kelompok lainnya memiliki karakteristik yang berbeda. Adapun tahapan-tahapan dalam K-Means Clustering adalah sebagai berikut :

1. Menentukan Jumlah k (cluster): Digunakan untuk acuan pengelompokan data.
2. Inisialisasi Centroid: Menentukan nilai centroid awal dipilih secara acak.
3. Pengelompokan Data: Setiap data diberikan label cluster berdasarkan jarak Euclidean terdekatnya ke salah satu centroid. Euclidean Distance adalah metode umum dalam mengukur jarak antara data point dan centroid. Data akan menjadi bagian dari kelompok yang memiliki centroid terdekat.

4. Perhitungan Centroid Baru: Setelah semua data dikelompokkan, centroid baru untuk setiap kelompok dihitung sebagai rata-rata dari data dalam kelompok tersebut.
5. Iterasi: Langkah 3 dan 4 diulangi sampai tidak ada perubahan dalam pengelompokan data atau jumlah iterasi yang ditentukan telah tercapai.
6. Hasil Akhir: Hasil akhir dari proses ini adalah pengelompokan data yang stabil, di mana data dalam setiap kelompok memiliki karakteristik yang mirip satu sama lain.

No	Programmer	Kriteria Penilaian				
		1	2	3	4	5
13	Programmer 13	1	4	2	3	3
14	Programmer 14	4	1	2	1	3
15	Programmer 15	3	4	4	1	1
16	Programmer 16	1	2	1	2	4
17	Programmer 17	1	3	4	4	3
18	Programmer 18	1	4	3	1	4
19	Programmer 19	3	1	1	2	1
20	Programmer 20	2	3	4	3	1

Tabel 3. Keterangan kriteria penilaian

Kriteria	
1	Kualitas Kode
2	Produktivitas
3	Kemampuan Teknis
4	Kolaborasi Tim
5	Kemampuan Pemecahan Masalah

Euclidean Distance

Euclidean Distance (Jarak Euclidean) adalah metode yang umum digunakan untuk mengukur jarak antara dua titik dalam ruang berdimensi. Metode ini berasal dari teorema Pythagoras dan diterapkan dalam berbagai konteks, termasuk dalam algoritma K-Means Clustering. Jarak Euclidean mengukur panjang garis lurus (jarak terpendek) antara dua titik dalam ruang multidimensi.

Tabel 4. Skala penilaian

Nilai	
1	Kurang
2	Cukup
3	Baik
4	Sangat Baik

HASIL DAN PEMBAHASAN

Pada penelitian ini, penulis melakukan clustering dengan 5 variabel :

Tabel 1. Variabel perhitungan

Variabel
Kualitas Kode
Produktivitas
Kemampuan Teknis
Kolaborasi Tim
Kemampuan Pemecahan Masalah

Tabel 2. Data kinerja programmer

No	Programmer	Kriteria Penilaian				
		1	2	3	4	5
1	Programmer 1	1	3	3	4	3
2	Programmer 2	4	4	4	3	4
3	Programmer 3	2	4	1	4	1
4	Programmer 4	1	3	2	3	2
5	Programmer 5	4	1	4	4	2
6	Programmer 6	4	2	2	1	4
7	Programmer 7	4	2	4	3	2
8	Programmer 8	4	2	4	1	2
9	Programmer 9	2	2	3	1	4
10	Programmer 10	4	1	4	1	3
11	Programmer 11	2	2	1	2	4
12	Programmer 12	3	1	2	2	4

1. Menentukan Jumlah Cluster: Penelitian ini menggunakan 3 cluster yang dipilih secara acak dengan variabel nilai kualitas kode, produktivitas, kemampuan teknis, kolaborasi tim, dan kemampuan pemecahan masalah.
2. Menentukan nilai tengah (centroid) awal: Untuk menentukan centroid awal dilakukan dengan menggunakan nilai random. Detail data centroid awal dapat dilihat pada Tabel 5 yang terdiri dari Cluster, Kualitas Kode (P1), Produktivitas (P2), Kemampuan Teknis (P3), Kolaborasi Tim (P4), dan Kemampuan Pemecahan Masalah (P5).

Tabel 5. Nilai centroid awal

Cluster	P1	P2	P3	P4	P5
C1	1	3	3	4	3
C2	1	4	3	1	4
C3	4	4	4	3	4

3. Menghitung jarak setiap data ke centroid dengan Euclidean.
4. Mengelompokkan data ke cluster terdekat.

Hasil perhitungan jarak setiap data dengan centroid awal dapat dilihat pada tabel 6.

Tabel 6. hasil perhitungan jarak iterasi 1

No	C1	C2	C3	Clus ter
0	0.000000	3.316625	3.605551	0
1	3.605551	3.741657	0.000000	2
2	3.162278	4.795832	4.795832	0
3	1.732051	3.162278	4.242641	0
4	3.872983	5.656854	3.741657	2
5	4.582576	3.741657	3.464102	2
6	3.605551	4.690416	2.828427	2
7	4.582576	4.242641	3.464102	2
8	3.464102	2.236068	3.605551	1
9	4.795832	4.472136	3.741657	2
10	3.316625	3.162278	4.242641	1
11	3.741657	3.872983	3.872983	0
12	1.732051	2.449490	3.741657	0
13	4.795832	4.472136	4.242641	2
14	4.358899	3.741657	3.741657	1
15	3.162278	3.000000	4.795832	1
16	1.000000	3.464102	3.464102	0
17	3.316625	0.000000	3.741657	1
18	4.472136	5.196152	5.385165	0
19	2.645751	4.000000	3.741657	0

Tabel 7. Tabel hasil pengelompokkan iterasi 1

Cluster	Jumlah
0	8
1	5
2	7

5. Menentukan nilai tengah (centroid) baru.

Setelah data dikelompokkan, selanjutnya perlu mencari centroid baru yang dihitung berdasarkan nilai rata-rata dari setiap variabel pada masing-masing cluster. Sehingga didapatkan nilai centroid baru sebagai berikut :

Tabel 8. Nilai centroid baru

Cluster	P1	P2	P3	P4	P5
C1	1.75	2.75	2.37	3.12	2.25
C2	1.80	2.80	2.40	1.40	3.40
C3	4.00	1.86	3.43	2.00	2.86

Langkah selanjutnya hitung kembali jarak setiap data ke centroid baru dengan menggunakan rumus Euclidean Distance seperti yang dilakukan pada langkah ke-2.

Tabel 9. Tabel hasil perhitungan jarak iterasi 2

No	C1	C2	C3	Clus ter
0	1.530931	2.821347	3.809226	0
1	3.513367	3.429286	2.687841	2
2	2.417385	3.994997	4.683885	0
3	0.918559	2.315167	3.752550	0
4	3.404960	4.400000	2.407471	2
5	3.652910	2.481935	2.089820	2
6	2.888555	3.544009	1.442786	2
7	3.583818	3.187475	1.442786	2
8	2.931510	1.249000	2.551510	1
9	3.980421	3.310589	1.442786	2
10	2.616056	1.833030	3.350297	1
11	3.015585	2.357965	2.254248	2
12	1.686342	2.227106	4.080816	0
13	3.652910	2.925748	1.948312	2
14	3.441475	3.370460	3.219836	2
15	2.709935	1.989975	4.027963	1
16	2.143304	3.187475	3.827932	0
17	3.177066	1.720465	4.010191	1
18	3.056755	3.572114	3.328909	0
19	2.084167	3.310589	3.175159	0

Tabel 10. Tabel hasil pengelompokkan iterasi 2

Cluster	Jumlah
1	7
2	4
3	9

Tabel 11. Tabel hasil perhitungan jarak iterasi 3

No	C1	C2	C3	Clus ter
0	1.470804	2.958040	3.651484	0
1	3.667285	3.840573	2.687419	2
2	2.175935	4.330127	4.496913	0
3	0.769309	2.598076	3.511885	0
4	3.588530	4.769696	2.560382	2
5	4.038084	2.598076	2.027588	2
6	3.073919	4.092676	1.527525	2
7	3.819926	3.840573	1.374369	2
8	3.276384	1.322876	2.357023	1
9	4.311825	3.708099	1.527525	2
10	2.979522	1.322876	3.179797	1
11	3.446382	2.179449	2.211083	1
12	1.609918	2.397916	3.844188	0
13	4.038084	3.122499	1.914854	2
14	3.425594	4.213075	3.000000	2
15	3.003399	1.322876	3.829708	1
16	2.075120	3.427827	3.696846	0
17	3.298113	1.936492	3.756476	1
18	3.276384	3.840573	3.197221	2
19	1.932536	3.968627	3.000000	0

Setelah melalui proses 3 kali iterasi didapat hasil akhir dengan total 6 orang pada cluster 1, 5 orang pada cluster 2, dan 9 orang pada cluster 3. Berikut tabel rekapitulasi hasil analisa perhitungan kinerja programmer dengan menggunakan K-Means Clustering.

Tabel 12. Tabel hasil pengelompokkan iterasi 3

Cluster	Jumlah
1	6
2	5
3	9

Tabel 13. Tabel rekapitulasi hasil

Cluster	Anggota Cluster	Jumlah
1	{0,2,3,8,10,12}	6
2	{8,10,11,15,16}	5
3	{1,4,5,6,7,9,13,14,17}	9

KESIMPULAN

Berdasarkan hasil analisis menggunakan metode K-Means Clustering pada data kinerja programmer di perusahaan software house, diperoleh tiga kelompok cluster yang memiliki karakteristik berbeda. Kelompok cluster 1 memiliki 6 anggota, kelompok cluster 2 memiliki 5 anggota, dan kelompok cluster 3 memiliki 9 anggota. Hasil ini menggambarkan variasi dalam kualitas kinerja programmer, yang dapat membantu perusahaan dalam mengidentifikasi potensi dan area yang perlu perbaikan dalam manajemen sumber daya manusia. Untuk penelitian selanjutnya, disarankan untuk memperluas data sampel dan mempertimbangkan faktor-faktor tambahan yang dapat mempengaruhi kualitas kinerja programmer, seperti pengalaman kerja atau tingkat pendidikan, guna memperoleh wawasan yang lebih komprehensif.

DAFTAR PUSTAKA

Amirulloh, I. (2019) ‘Pemetaan Kelompok Kerja Siswa dengan Metode Clustering K-Means dan Algoritma Greedy’, *Jurnal Informatika dan Rekayasa Perangkat Lunak*, 1(2), pp. 94–98. doi:10.36499/jinrpl.v1i2.2953.

Mauludin, M.S. and Suali (2019) ‘Sistem Pendukung Keputusan Penilaian Kinerja Karyawan Menggunakan Metode TOPSIS di PT Karya Mitra Nugraha’, *Informatika dan RPL*, 1(2), pp. 88–93.

Nurahman and Dwi Aulia, D. (2023)

‘Klasterisasi Pendidikan Masyarakat untuk mengetahui Daerah dengan Pendidikan Terendah menggunakan Algoritma K-Means’, *Jurnal Informatika dan Rekayasa Perangkat Lunak*, 5(1), pp. 38–44.

Saputra, M.A.W. et al. (2021) ‘Survei Teknik-Teknik Pengukuran Kualitas Perangkat Lunak’, *ILKOMNIKA: Journal of Computer Science and Applied Informatics*, 3(1), pp. 11–29. doi:10.28926/ilkomnika.v3i1.38.

Sutisna, G. (2019) ‘Analisis Teamwork Karyawan di PT Nusantara Cemerlang’, *Prosiding FRIMA (Festival Riset Ilmiah Manajemen dan Akuntansi)*, (2), pp. 975–979. doi:10.55916/frima.v0i2.131.

Syahputra, M.A.D. (2021) ‘Pentingnya Pendekatan Interpersonal Skills untuk Mengembangkan Hard Skill & Soft Skill pada Mahasiswa’, *JEJAK: Jurnal Pendidikan Sejarah & Sejarah*, 1(2), pp. 82–90. doi:10.22437/jejak.v1i2.16320.

Thaher, I.A., Septiarini, A. and Puspitasari, N. (2022) ‘Pengelompokan Kualitas Kinerja Pegawai Menggunakan Metode K-Means Clustering’, *Komputika: Jurnal Sistem Komputer*, 11(2), pp. 131–141. doi:10.34010/komputika.v11i2.5518.

Toharudin, U. and Setia Kurniawan, I. (2017) ‘Studi Kasus Pada Psikologi Pendidikan: Bagaimana Mahasiswa Memiliki Kemampuan Problem Solving’, *Jurnal Pendidikan Biologi Indonesia*, 3(1), pp. 36–44. Available at: <http://ejournal.umm.ac.id/index.php/jpb> i.

Wahyuningsih, S. (2019) ‘Pengaruh Pelatihan dalam Meningkatkan Produktivitas Kerja Karyawan’, *Jurnal Warta Edisi: 60*, (April), pp. 91–96.