

Optimasi *K-Means* untuk Pengelompokan Data Kinerja Akademik Dosen menggunakan *Particle Swarm Optimization*

Dian Werdiningsih Dwi Rahmawati¹, Imam Cholissodin², Nurudin Santoso³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹dian.werdiningsih18@gmail.com, ²imamcs@ub.ac.id, ³nurudin.santoso@ub.ac.id

Abstrak

Dosen merupakan pengajar dan pembimbing bagi mahasiswa. Selain mengajar dan membimbing, dosen juga diharuskan membantu dalam mengembangkan ilmu pengetahuan serta potensi maupun keahlian pada dirinya sendiri. Variabel atau kriteria yang ada pada dosen meliputi Pendidikan, penelitian, pengabdian, administrasi, dan penunjang. Salah satu kesulitan yang dihadapi pada perguruan tinggi adalah pengelompokan yang ideal penugasan terhadap dosen. Penugasan dosen terkait dengan kepanitiaan, *study* lanjut, jabatan, pengisi suatu acara yang diselenggarakan dari internal maupun eksternal perguruan tinggi dan lain-lain. Sehingga dibutuhkan sistem yang dapat mengelompokkan kinerja akademik dosen menggunakan algoritma *K-Means* yang dipadukan dengan algoritma *Particle Swarm Optimization*. Algoritma *Particle Swarm Optimization* berguna dalam mengoptimalkan *centroid* pada algoritma *K-Means*. Data yang digunakan merupakan data dari Fakultas Ilmu Komputer (FILKOM), Universitas Brawijaya tahun 2016. Data tersebut diperoleh dari GJM FILKOM. Hasil pengujian dalam penelitian ini memperoleh algoritma *K-Means* yang dipadukan dengan algoritma *Particle Swarm Optimization* lebih baik 3.28% dibandingkan dengan algoritma *K-Means* yang berdiri sendiri, dimana kualitas kluster ditentukan menggunakan metode *Silhouette Coefficient*.

Kata kunci: *Particle Swarm Optimization, K-Means, Silhouette Coefficient, Kinerja Akademik Dosen.*

Abstract

Lecturers are teachers and tutors for students. In addition to teaching and guiding, lecturers are also required to assist in developing science as well as potential and expertise in itself. Variables or criteria that exist in the lecturer include Education, research, devotion, administration, and support. One of the difficulties faced by universities is the ideal grouping of assignments to lecturers. Assignment of lecturers related to the committee, further study, positions, filler of an event held from internal and external universities and others. So it takes a system that can classify the academic performance of lecturers using *K-Means* algorithm combined with *Particle Swarm Optimization* algorithm. The *Particle Swarm Optimization* algorithm is useful in optimizing the *centroid* on the *K-Means* algorithm. The data used is data from the Faculty of Computer Science (FILKOM), Universitas Brawijaya 2016. The data is obtained from GJM FILKOM. The results of this study obtained *K-Means* algorithm combined with *Particle Swarm Optimization* algorithm better 3.28% compared with stand-alone *K-Means* algorithm, where the cluster quality is determined using *Silhouette Coefficient* method.

Keywords: *Particle Swarm Optimization, K-Means, Silhouette Coefficient, Academic Performance of Lecturers.*

1. PENDAHULUAN

Dosen adalah seorang pengajar untuk mahasiswa di perguruan tinggi. Dosen juga membimbing mahasiswanya agar mahasiswa tersebut memiliki kompetensi yang baik dalam keahliannya. Selain mengajar dan membimbing, dosen juga melakukan tugas lain dalam mengembangkan ilmu pengetahuan serta

mengembangkan potensi diri.

Penugasan dosen ditentukan dari hasil kinerja dosen tersebut. Begitu juga kualitas suatu perguruan tinggi dapat dilihat dari kinerja dosen yang dimiliki perguruan tinggi itu sendiri. Dimana kinerja dosen tersebut dapat dijadikan tolok ukur dalam melakukan evaluasi. Evaluasi tersebut bertujuan untuk mencapai visi dan misi perguruan tinggi, meningkatkan mutu dalam

pembelajaran dan pelayanan pendidikan, dan untuk akreditasi.

Bila kinerja dosen tidak dievaluasi secara teratur, maka akan berdampak buruk bagi perguruan tinggi, dan mahasiswa. Secara umum dampak yang terjadi pada mahasiswanya adalah kurangnya ilmu pengetahuan yang didapatkan sehingga bila mahasiswa tersebut lulus akan susah dalam mencari pekerjaan, sulit dalam memecahkan masalah yang dihadapi karena kurangnya bimbingan, dan lain-lain. Kemudian dampak buruk untuk perguruan tinggi adalah menurunkan akreditasi, pamor perguruan tinggi akan menurun sehingga akan menurunkan peminat yang ingin masuk ke perguruan tinggi tersebut, susah bersaing dengan perguruan tinggi lainnya, dan lain-lain.

Kesulitan yang dialami pada perguruan tinggi salah satunya dalam menentukan penugasan bagi dosennya. Sehingga tujuan dalam penelitian ini adalah bagaimana nilai kualitas *cluster* yang dihasilkan algoritma *K-Means* yang dipadukan dengan algoritma *Particle Swarm Optimization* dalam mengelompokkan kinerja akademik dosen. Sehingga dapat membantu mengevaluasi dosen dan tidak terjadi simpang-siur dalam pemberian penugasan kepada dosen.

Clustering merupakan metode pengelompokan data ke dalam beberapa *cluster*, sehingga data pada *cluster* yang sama memiliki tingkat kemiripan yang tinggi dan pada *cluster* yang lain memiliki kemiripan yang rendah (Tan, 2006). Algoritma dalam *Clustering* sangatlah banyak salah satunya adalah algoritma *K-Means*.

Algoritma *K-Means* merupakan algoritma yang sederhana dalam mengelompokkan data. Selain sederhana, algoritma *K-Means* sangat efisien serta dapat digunakan dalam memproses data yang berukuran besar (Tan, Steinbech, & Kumar, 2006). *K-Means* memiliki kemampuan yang kuat dalam pencarian lokal yang optimal, akan tetapi dalam pencarian global yang optimal sangatlah lemah. Sehingga perlu metode yang dapat membantu mengoptimalkan algoritma *K-Means*.

Particle Swarm Optimization merupakan sebuah teknik yang dapat digunakan untuk meningkatkan *centroid K-Means* dengan hasil yang lebih baik (Kuo et al, 2011). *PSO* juga memiliki kemampuan yang kuat dalam pencarian hasil global yang optimal.

Salah satu penelitian yang menggunakan algoritma *K-Means* yang dipadukan dengan algoritma *Particle Swarm Optimization* adalah

Esmin dan Matwin (2012) yang menerapkan algoritma *PSO* terlebih dahulu untuk mencari pusat cluster kemudian pusat cluster tersebut digunakan dalam algoritma *K-Means* untuk beberapa objek.

Sehingga penulis membuat judul "Optimasi *K-Means* Untuk Pengelompokan Kinerja Akademik Dosen Menggunakan *Particle Swarm Optimization*" sebagai topik skripsi yang diambil. Penelitian ini menggabungkan algoritma *Particle Swarm Optimization* dengan *K-Means* dalam mengelompokkan kinerja akademik dosen. Titik pusat klaster akan dioptimasi terlebih dahulu dalam proses *PSO*. Kemudian setelah itu, titik pusat klaster tersebut akan menjadi titik pusat awal dalam proses *K-Means*. Selanjutnya akan dihitung kualitas klaster dengan *Silhouette Coefficient*.

2. KINERJA DOSEN

Kinerja dosen adalah sebuah rekam jejak dari seorang dosen, dimana seorang dosen menunjukkan kemampuannya dalam menyelesaikan sebuah tugas sesuai tanggungjawab yang diberikan.

Pengaruh Kinerja dosen dapat dilihat dari beberapa variable atau kriteria seperti pendidikan, penelitian, pengabdian, administrasi, dan penunjang. Dalam variable pendidikan meliputi jumlah pertemuan kuliah, hasil evaluasi perkuliahan, jumlah sks, dan lain-lain. Selanjutnya dalam variable penelitian meliputi jumlah penelitian tiap tahun, jurnal nasional maupun internasional tiap tahun, menghasilkan karya sesuai bidang keahlian, dan lain-lain. Dalam variabel pengabdian meliputi melaksanakan pengabdian tiap tahun, memiliki publikasi karya pengabdian, dan lain-lain. Dalam variable administrasi meliputi jabatan akademik, Pendidikan atau *study*, mengikuti pelatihan, dan lain-lain. Terakhir dalam variable penunjang meliputi terlibat dalam kepanitiaan, aktif dalam pertemuan ilmiah, dan lain-lain.

3. CLUSTERING

Clustering adalah proses pengelompokan data menjadi beberapa bagian sehingga data dalam suatu *cluster* memiliki kesamaan yang tinggi, namun jauh berbeda dengan data dalam *cluster* lain (Jiawei Han, et all, 2012:443).

Secara umum metode *clustering* dapat dikelompokkan ke dalam beberapa kategori seperti (Jiawei Han, et all, 2012:448) :

1. Metode partisi.
Metode yang membagi data menjadi beberapa kelompok sehingga setiap kelompok setidaknya mengandung satu objek. Dengan kata lain metode ini melakukan partisi satu tingkat pada kumpulan data dan pada dasarnya mengadopsi pemisahan kelompok secara eksklusif atau setiap objek harus dimiliki oleh satu kelompok.
2. Metode hirarki
Metode yang membentuk dekomposisi hirarki dari sekumpulan data dan dapat diklasifikasikan sebagai aglomeratif atau memecah belah berdasarkan dekomposisi hirarki yang terbentuk. Aglomeratif ini bisa disebut juga dengan pendekatan bottom-up (awalnya setiap objek membentuk kelompok yang terpisah, kemudian menggabungkan objek atau kelompok yang berdekatan satu sama lain hingga menjadi satu) dan top-down (diawali dengan tiap objek berada dalam cluster yang sama, kemudian *cluster* dibagi dalam kelompok yang lebih kecil hingga tiap objek berada dalam satu cluster).
3. Metode berbasis kepadatan
Metode ini dapat membagi satu set objek menjadi beberapa kelompok yang eksklusif. Biasanya metode ini hanya mempertimbangkan kelompok yang eksklusif saja dan tidak menganggap kelompok *fuzzy*.
4. Metode berbasis grid
Metode ini menguatifikasi ruang objek menjadi sejumlah cell yang terbatas yang membentuk struktur grid. Kemudian semua operasi pengelompokan dilakukan pada struktur grid yaitu pada ruang terkuantisasi.

4. ALGORITMA K-MEANS

Algoritma *K-Means* merupakan salah satu algoritma untuk mengelompokkan beberapa data, Dimana *K-Means* termasuk metode *clustering* non-hirarki yang akan mengelompokkan data dalam beberapa kelompok yang memiliki kesamaan atau karakteristik yang sama untuk menjadi satu kelompok, dan bila berbeda karakteristik maka disatukan dalam kelompok lain, sehingga data pada satu kelompok mempunyai tingkat variasi yang sangat minim atau sedikit (Agusta, 2007).

Langkah – Langkah Algoritma *K-Means* (Junjie Wu, 2012:7) :

1. Tentukan *K cluster centroid* yang diinginkan.

2. Hitung jarak setiap data dengan *centroid* dengan Persamaan (1) dimana N_x adalah jumlah data, N_f adalah jumlah dimensi, x_{ij} adalah data ke-I dengan atribut ke-j, dan c_{ij} adalah kluster ke-I dengan *centroid* ke-j.

$$d(x_{ij}, c_{ij}) = \sqrt{\sum_{i=1}^{N_x} \sum_{j=1}^{N_f} (x_{ij} - c_{ij})^2} \quad (1)$$

3. Setiap data dikelompokkan berdasarkan jarak terdekat dengan *centroid*
4. Perbarui *centroid* berdasarkan data yang telah dikelompokkan dengan Persamaan (2) dimana x_j adalah atribut ke-j pada suatu data, N_d adalah jumlah dimensi, dan N_{xc} adalah jumlah data dalam satu kluster.

$$C_j = \frac{\sum_{j=1}^{N_d} x_j}{N_{xc}} \quad (2)$$

5. Ulangi hingga tidak terjadi perubahan pada nilai *centroid*, dan nilai jarak minimal maupun maksimal data ke *cluster* kurang dari *threshold*.

5. ALGORITMA PSO

PSO berawal dari bentuk sekelompok burung, dimana setiap partikel diikuti dengan perilaku yang sederhana yaitu mengikuti partikel yang terbaik dan bergerak menuju kondisi terbaik yang ditemukan oleh partikel itu sendiri (Konstantinos dan Michael, 2010).

Langkah –langkah algoritma *PSO* (Venkata S.P, 2009) :

1. Inisialisasi jumlah partikel secara acak
2. Hitung fitness dari setiap partikel
3. Perbarui nilai *Pbest* dengan Persamaan (3) dimana $F(x_i)$ adalah nilai fitness pada partikel ke-i, $F(Pbest_i)$ adalah nilai fitness pada partikel baik ke-i.

$$Pbest_i = F(x_i) \geq F(Pbest_i) \quad (3)$$

4. Perbarui nilai *Gbest* dengan persamaan (4) dimana $MAX(F(Pbest_1), \dots, F(Pbest_n))$ adalah fitness terbesar dari seluruh *Pbest*
 $Gbest_i = MAX (F(Pbest_1), \dots, F(Pbest_n))$ (4)
5. Perbarui nilai kecepatan dengan persamaan (5) dimana w adalah bobot inersia, $v_{ij}(t)$ adalah kecepatan pada iterasi ke-t, c_1 dan c_2 adalah nilai konstanta akselerasi, r_1 dan r_2 adalah nilai random dari 0 sampai dengan 1, $x_{ij}(t)$ adalah partikel ke-t dengan atribut ke-j pada iterasi ke-t, $P_{ij}(t)$ adalah partikel baik ke-i dengan atribut ke-j, dan $G_j(t)$ adalah atribut ke-j pada partikel terbaik.

$$v_{ij}(t + 1) = w \cdot v_{ij}(t) + c_1 \cdot r_1 [P_{ij}(t) - x_{ij}(t)] + c_2 \cdot r_2 [G_j(t) - x_{ij}(t)] \quad (5)$$

- Perbarui nilai posisi dengan persamaan (6) dimana $v_{ij}(t+1)$ adalah nilai kecepatan pada iterasi ke- $t+1$, dan $x_{ij}(t)$ adalah partikel ke- i dengan atribut ke- j pada iterasi ke- t .

$$x_{ij}(t + 1) = x_{ij}(t) + v_{ij}(t + 1) \quad (6)$$

- Ulangi langkah 2, 3, dan 4 hingga iterasi maksimum.

6. NILAI FITNESS PSO

Nilai Fitness algoritma *PSO* didapatkan dari jarak rata-rata antara data dan *cluster* digunakan untuk mengevaluasi setiap solusi yang ditunjukkan oleh partikel. Nilai fitness diukur dengan persamaan di bawah ini (Xiaohui Cui, et al, 2005). Menghitung fitness menggunakan persamaan (7) dimana N_c adalah jumlah kluster, N_x adalah jumlah data pada kluster ke- i , dan $d(x_j, c_i)$ adalah jarak antara data ke- i dengan kluster ke- i .

$$f = \frac{\sum_{i=1}^{N_c} \left(\frac{\sum_{j=1}^{N_x} d(x_j, c_i)}{N_x} \right)}{N_c} \quad (7)$$

7. SILHOUETTE COEFFICIENT

Silhouette Coefficient merupakan sebuah teknik yang dapat digunakan dalam mengukur seberapa baik suatu objek atau data dalam suatu *cluster* (Anggara et al, 2016). Berikut tahapan dalam perhitungan *Silhouette Coefficient* :

- Hitung rata - rata jarak pada data ke - i dengan semua data yang berada pada satu *cluster* dengan menggunakan persamaan (8) dimana $d(x_i, x'_i)$ adalah jarak antara data x_i dengan data x'_i dalam satu kluster c_i dimana $x_i \neq x'_i$, dan N_{xci} adalah jumlah data dalam satu kluster.

$$a_i = \frac{\sum_{\forall x_i \in c_i, x_i \neq x'_i} d(x_i, x'_i)}{N_{xci}} \quad (8)$$

- Hitung rata - rata jarak pada data ke - i dengan semua data yang terletak pada *cluster* yang berbeda dengan data ke - i tersebut, kemudian ambil nilai yang terkecil atau minimum. Pada persamaan (9) dimana $d(x_i, x'_i)$ adalah jarak antara data x_i dengan data x'_i dalam satu kluster c_i dimana $x_i \neq x'_i$, dan N_{xci} adalah jumlah data yang termasuk dalam kluster selain c_i . pada persamaan (10) b_i adalah nilai minimum dari rata-rata jarak antara kluster c'_i dengan kluster c''_i .

$$d(x_i, c_i) = \frac{\sum_{\forall x_i \in c_i, x_i \neq x'_i} d(x_i, x'_i)}{N_{xci}} \quad (9)$$

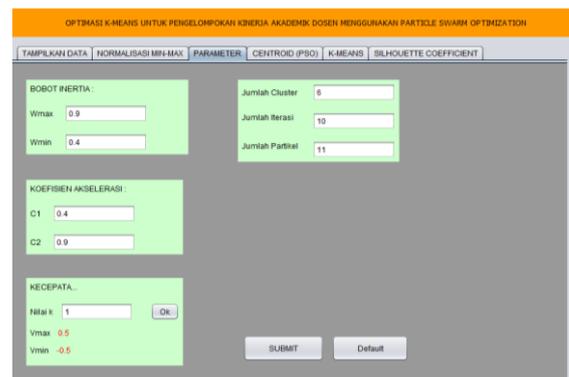
$$b_i = \text{MIN } c'_i \neq c''_i d(x_i, c_i) \quad (10)$$

- Hitung nilai *Silhouette Coefficient* untuk seluruh *centroid* menggunakan persamaan (11) dimana a_i adalah nilai rata-rata jarak data ke- i dengan data lain dalam satu kluster, b_i adalah nilai terkecil dari rata-rata jarak antara data ke- i dengan data lain dalam beda kluster.

$$S_i = (b_i - a_i) / \text{MAX}(a_i, b_i) \quad (11)$$

8. IMPLEMENTASI

Pada proses implementasi ini partikel terbaik yang dihasilkan pada algoritma *PSO* akan menjadi titi pusat kluster awal untuk proses algoritma *K-Means* dalam melakukan proses pengelompokan. Selanjutnya hasil pengelompokan dari algoritma *K-Means* akan dihitung nilai kualitas kluster dalam proses *Silhouette Coefficient*.



Gambar 8.1 Halaman Parameter

Pada gambar 5.3 user akan memasukan parameter yang dibutuhkan dalam algoritma *PSO* dan *K-Means*. Disamping itu juga jika *user* ingin mengubah parameter kecepatan bisa memasukan nilai k terlebih dahulu dan menekan tombol *ok* untuk melihat hasil v_{max} dan v_{min} . Setelah *user* telah memastikan parameter terisi dengan benar, *user* bisa langsung menekan tombol *submit* untuk memulai perhitungan pada sistem.

Cluster	Pendidikan	Penelitian	Pengabdian	Administrasi	Penunjang
1	0.45782	0.33333	0	0	1
2	0.41896	0.44444	0	0.66667	1
3	0.34814	0.65556	0	1	0
4	0.05793	0.55556	1	0	0
5	0.11154	0.22222	0	0	0
6	0.21976	0.22222	0	0	0

Gambar 8.2 Halaman Centroid(PSO)

Pada gambar 5.4 pengguna dapat melihat hasil titik pusat kluster yang telah dioptimasi oleh algoritma *PSO* dalam bentuk tabel.

OPTIMASI K-MEANS UNTUK PENGELOMPOKAN KIBERA AKADEMIK DOSEN MENGGUNAKAN PARTICLE SWARM OPTIMIZATION						
TAMPILKAN DATA	NORMALISASI MIN-MAX	PARAMETER	CENTROID (PSO)	K-MEANS	SILHOUETTE COEFFICIENT	
Dosen	Pendidikan	Penelitian	Pengabdian	Administrasi	Penunjang	Cluster
1	0.33644	0.55556	0	0	1	1
2	0	0.33333	0	0.33333	0	5
3	0.52755	1	0	0.66667	1	2
4	0.38163	0.33333	0	0.33333	1	1
5	0.21976	0.22222	0	0	0	6
6	0.67663	0.55556	1	0	0	4
7	0.11154	0.22222	0	0	0	5
8	0.41337	0.22222	0	0.33333	0	6
9	1	0.55556	0	0.33333	0	5
10	0.46782	0.33333	0	0	1	1
11	0.12733	0.44444	0	0.33333	0	5
12	0.48223	0.33333	1	0	1	1
13	0.30287	0	0	0	0	6
14	0.07267	0.22222	0	0	0	5
15	0.72843	0.44444	0	0	1	1
16	0.18635	0.22222	0	0	0	6
17	0.14019	0.11111	0	0	1	1
18	0.63835	0.22222	0	0.66667	1	2
19	0.41895	0.44444	0	0.66667	1	2
20	0.69074	0.55556	0	0.66667	1	2
21	0.38559	0.11111	0	0	1	1
22	0.01565	0.11111	0	0	1	1
23	0.34614	0.55556	0	1	0	3
24	0.65349	0.44444	0	0	1	1
25	0.58215	0.11111	1	0	1	1

Gambar 8.3 Halaman K-Means

Pada gambar 5.5 pengguna dapat melihat hasil pengelompokan data dari proses algoritma *K-Means* dalam bentuk tabel.

OPTIMASI K-MEANS UNTUK PENGELOMPOKAN KIBERA AKADEMIK DOSEN MENGGUNAKAN PARTICLE SWARM OPTIMIZATION					
TAMPILKAN DATA	NORMALISASI MIN-MAX	PARAMETER	CENTROID (PSO)	K-MEANS	SILHOUETTE COEFFICIENT
Dosen	ai	bi	max(ai,bi)	si	
1	0.50407	0.76593	0.76593	0.34189	
2	0.22411	0.56119	0.56119	0.80207	
3	0.45629	1.06863	1.06863	0.57301	
4	0.52496	0.5135	0.52496	-0.02182	
5	0.17755	0.27023	0.27023	-0.14902	
6	0	1.06267	1.06267	1	
7	0.20211	0.38235	0.38235	0.4714	
8	0.37613	0.42915	0.42915	0.19888	
9	0.69914	0.9338	0.9338	0.25129	
10	0.42653	0.76415	0.76415	0.4417	
11	0.24356	0.52764	0.52764	0.53839	
12	0.86246	1.10133	1.10133	0.21889	
13	0.37307	0.43903	0.43903	0.15025	
14	0.2005	0.41553	0.41553	0.51749	

Cluster	Fitness
1	0.32092
2	0.1424
3	1
4	1
5	0.53234
6	0.01356

RATA-RATA FITNESS SELURUH CLUSTER: 0.58018

KUALITAS CLUSTER: 0.3793583500635822

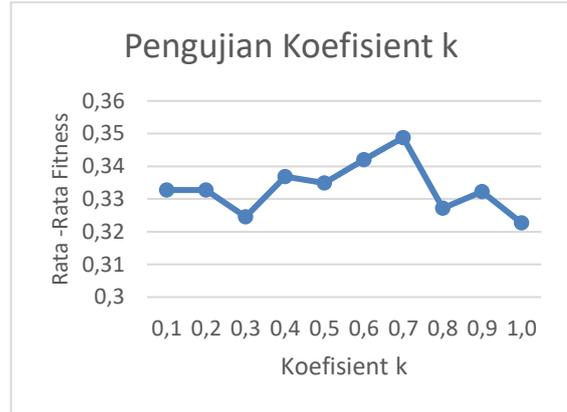
Gambar 8.4 Halaman Silhouette Coefficient

Pada gambar 5.6 pengguna bisa melihat hasil perhitungan ai, bi, max(ai,bi), si tiap data, nilai fitness tiap kluster dalam bentuk tabel. Pengguna juga dapat melihat Nilai rata-rata fitness kluster, dan nilai kualitas kluster yang dihasilkan dalam proses *Silhouette Coefficient*.

9. PENGUJIAN DAN ANALISIS

9.1. Pengujian Koefisien k

Proses ini bertujuan untuk membatasi nilai kecepatan yang akan digunakan dalam proses algoritma *PSO*. Dalam Proses ini menggunakan 10 partikel, $w_{max}=1$, $w_{min}=1$, $c_1=1$, $c_2=1$, jumlah kluster=3.

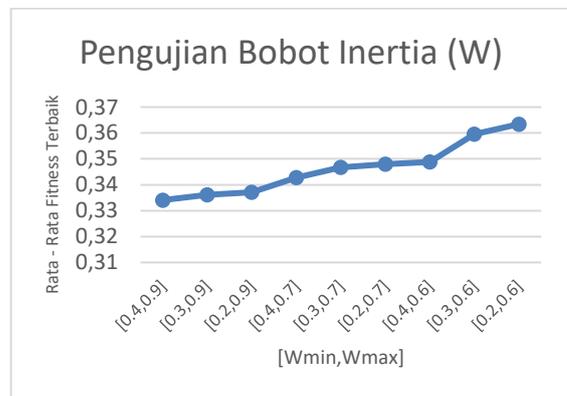


Gambar 9.1 Grafik Pengujian Koefisien k

Pada gambar 6.1 bisa dilihat bahwa saat nilai $k = 0.3$ hingga $k = 1.0$ terjadi perubahan yang sangat signifikan. Nilai rata-rata fitness terkecil terjadi saat nilai inputan $k = 1.0$ sedangkan Nilai rata-rata fitness terbesar terjadi saat nilai inputan $k = 0.7$ sehingga nilai k yang terbaik adalah bernilai 1.0.

9.2. Pengujian Bobot Inersia

Proses ini bertujuan untuk mengetahui nilai bobot inersia yang terbaik dalam proses algoritma *PSO*. Dalam proses ini menggunakan 10 partikel, nilai koefisien $k=1$, nilai $c_1=1$, nilai $c_2=1$, jumlah kluster=3.



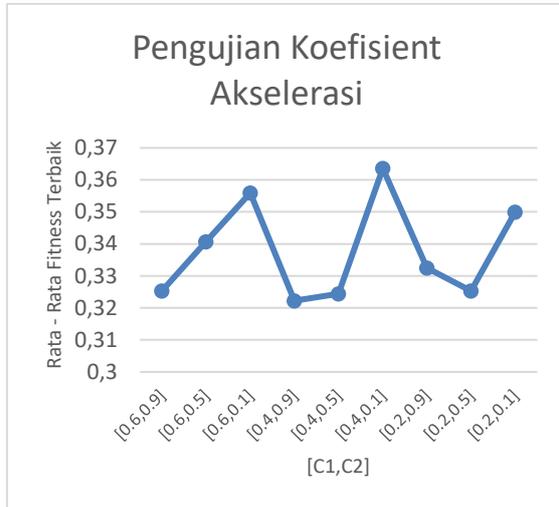
Gambar 9.2 Grafik Pengujian Bobot Inersia

Pada gambar 6.2 dapat dilihat bahwa perubahan nilai rata-rata fitness tidak terlalu jauh satu sama lainnya. Nilai rata-rata fitness tertinggi terdapat pada $w_{max} = 0.6$ dan $w_{min} = 0.2$ sedangkan nilai rata-rata fitness terendah pada nilai $w_{max}=0.9$ dan $w_{min}=0.4$ sehingga kombinasi keduanya merupakan nilai bobot inersia terbaik.

9.3. Pengujian Koefisien Akselerasi

Proses ini bertujuan untuk mengetahui kombinasi nilai c_1 dan nilai c_2 yang terbaik pada

proses algoritma *PSO*. Dalam Proses ini menggunakan 10 partikel, $w_{max}=0.9$, $w_{min}=0.4$, nilai koefisien $k=1$, jumlah klaster=3.

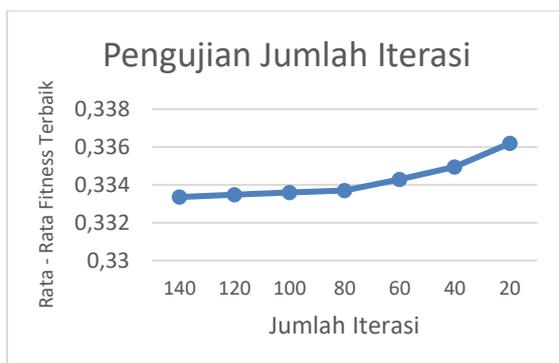


Gambar 9.3 Grafik Pengujian Koefisient Akselerasi

Pada gambar 6.3 dapat dilihat bahwa perubahan nilai rata-rata *fitness* terlihat signifikan. Saat nilai $c1=0.4$ dan $c2=0.9$ hingga $c1=0.4$ dan $c2=0.5$ perubahan rata-rata *fitness* tidak terlalu jauh. Dengan nilai $c2$ yang kecil cenderung menghasilkan nilai *fitness* yang tinggi. Nilai dengan rata-rata *fitness* tertinggi diperoleh oleh nilai $c1=0.4$ dan $c2=0.1$ sedangkan nilai dengan rata-rata terendah diperoleh oleh nilai $c1=0.4$ dan $c2=0.9$. Sehingga kombinasi nilai $c1=0.4$ dan $c2=0.9$ merupakan nilai koefisien akselerasi yang terbaik.

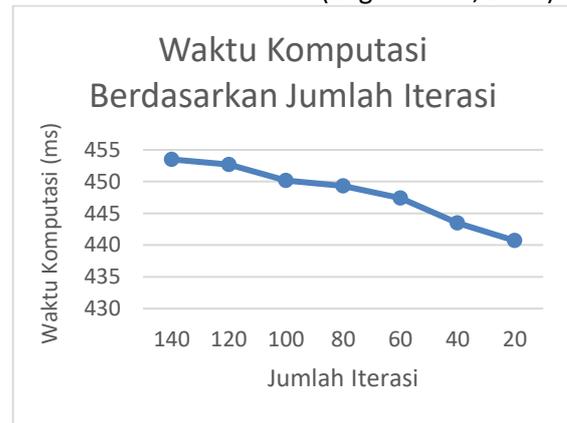
9.4. Pengujian Jumlah Iterasi

Proses ini bertujuan untuk mengetahui jumlah terasi yang terbaik pada proses algoritma *PSO*. Dalam Proses ini menggunakan 10 partikel, nilai $w_{max}=0.9$, nilai $w_{min}=0.4$, nilai koefisien $k=1$, nilai $c1=0.4$, nilai $c2=0.9$, dan jumlah klaster=3.



Gambar 9.4 Grafik Pengujian Jumlah Iterasi

Pada gambar 6.4 dapat dilihat bahwa nilai rata-rata *fitness* terendah diperoleh oleh iterasi ke 140. Jumlah iterasi yang diperlukan untuk mendapatkan solusi terbaik bergantung pada variasi sebuah permasalahan yang dihadapi. Jumlah iterasi yang terlalu sedikit akan mengakibatkan pencarian solusi menjadi tidak optimum. Sebaliknya jika iterasi terlalu banyak akan mengakibatkan waktu yang dihabiskan semakin lama (Engelbrecht, 2007).

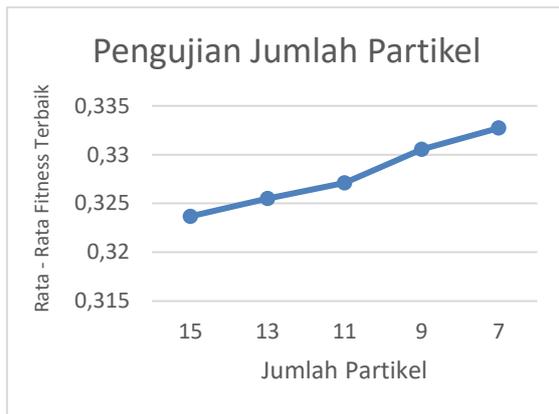


Gambar 9.5 Grafik Waktu Komputasi Pengujian Jumlah Iterasi

Pada gambar 6.5 terlihat bila jumlah iterasi semakin banyak maka waktu komputasi semakin meningkat juga. Oleh karena itu direkomendasikan memakai jumlah iterasi = 80 karena mulai iterasi = 80 nilai rata-rata *fitness* yang cukup rendah dan lebih efisien.

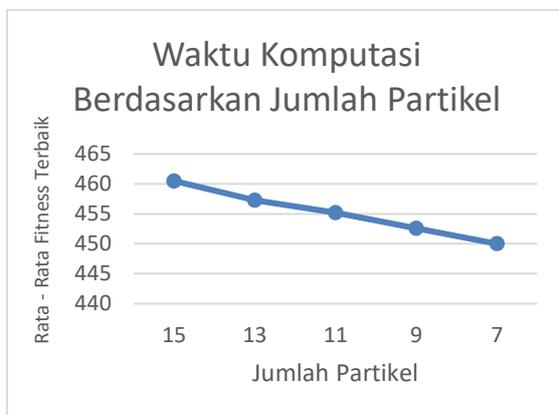
9.5. Pengujian Jumlah Partikel

Proses ini bertujuan untuk mengetahui jumlah partikel yang terbaik pada proses algoritma *PSO*. Dalam Proses ini menggunakan nilai $w_{max}=0.9$, nilai $w_{min}=0.4$, nilai koefisien $k=1$, nilai $c1=0.4$, nilai $c2=0.9$, jumlah klaster=3 dan jumlah iterasi=80.



Gambar 9.6 Grafik Pengujian Jumlah Partikel

Pada gambar 6.6 dapat dilihat bahwa dengan jumlah partikel = 15 dapat menghasilkan rata-rata *fitness* yang baik dikarenakan memiliki nilai paling kecil atau nilai *error* yang kecil. Semakin besar jumlah partikel maka ruang pencarian semakin luas sehingga peluang dalam menghasilkan solusi yang optimum semakin meningkat. Namun bila jumlah partikel semakin banyak maka waktu yang dikonsumsi juga semakin lama (Engelbrecht, 2007).



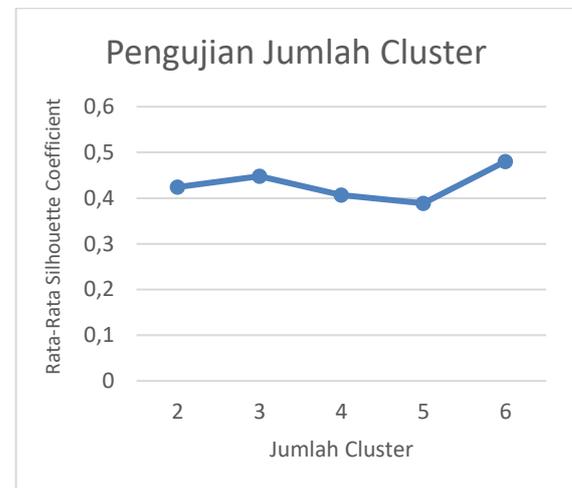
Gambar 9.7 Grafik Waktu Pengujian Partikel

Hal tersebut bisa dilihat pada gambar 6.7 bila semakin besar jumlah partikel maka waktu yang digunakan semakin tinggi. Oleh karena itu disarankan memilih jumlah partikel sebanyak 11 partikel karena memiliki rata-rata *fitness* yang cukup rendah dan membutuhkan waktu yang tidak terlalu lama.

9.6. Pengujian Jumlah Cluster

Proses ini bertujuan untuk mengetahui kualitas kluster yang baik dengan metode *Silhouette Coefficient*. Sehingga dari nilai

kualitas kluster tersebut bisa diketahui jumlah *Cluster* yang terbaik pada proses algoritma *K-Means*. Dalam Proses ini menggunakan 11 partikel, nilai $w_{max}=0.9$, nilai $w_{min}=0.4$, nilai koefisien $k=1$, nilai $c_1=0.4$, nilai $c_2=0.9$, dan jumlah iterasi=80.



Gambar 9.8 Grafik Pengujian Jumlah Cluster

Pada gambar 6.8 dapat dilihat bahwa dengan jumlah *cluster* = 6 dapat menghasilkan rata-rata *Silhouette Coefficient* yang baik. Hal tersebut menunjukkan bahwa dengan jumlah kluster = 6 maka tingkat kemiripan data dalam satu kluster yang sama semakin mirip dan tingkat kemiripan data dalam kluster yang berbeda semakin tidak mirip. Nilai *Silhouette Coefficient* berada pada interval -1 sampai 1. Ketika nilai *Silhouette Coefficient* mendekati 1, maka jarak antar data dalam satu kluster yang sama semakin dekat dan jarak data dengan kluster yang berbeda semakin jauh. Sebaliknya ketika nilai *Silhouette Coefficient* mendekati -1, maka jarak antar data dalam satu kluster semakin jauh dan jarak data dengan kluster yang berbeda semakin dekat (Han, et al., 2012).

9.7. Pengujian Perbandingan Algoritma

Proses ini bertujuan untuk mengetahui kinerja algoritma *K-Means* dan algoritma *PSO* dengan algoritma *K-Means* yang berdiri sendiri. Pada Table 6.37 dapat dilihat bahwa algoritma *HPSOKM* memiliki nilai *Silhouette Coefficient* lebih tinggi dibandingkan dengan algoritma *K-Means*. Hal tersebut menjelaskan bahwa algoritma *HPSOKM* menghasilkan pengelompokan yang lebih baik dari pada algoritma *K-Means* meskipun memakan waktu yang lama dengan peningkatan 3.28%. Sehingga

dapat disimpulkan bahwa algoritma *HPSOKM* memiliki kinerja yang lebih baik dibandingkan algoritma *K-Means* yang berdiri sendiri dalam pengelompokan data kinerja dosen.

Tabel 9.1 Hasil Pengujian Perbandingan Algoritma

	Waktu Komputasi (ms)	Interval -1 s.d 1
K-Means	406.9	0.38362882907171914
HPSOKM	514.6	0.4538770713791105

10. KESIMPULAN

Berdasarkan hasil pengujian dan analisis dari penerapan algoritma dalam mengoptimalkan *K-Means* menggunakan *PSO* pada data kinerja kademik dosen maka dapat ditarik kesimpulan sebagai berikut :

1. Penambahan *cluster* berpengaruh dalam kualitas hasil pengelompokan. Jumlah klaster terbaik berdasarkan pengujian adalah 6 *cluster* dengan nilai *Silhouette Coefficient* tertinggi yaitu 74.01%.
2. Algoritma optimasi *K-Means* dengan *PSO* terbukti lebih baik dalam melakukan pengelompokan data kinerja dosen dibandingkan dengan algoritma *K-Means* itu sendiri. Berdasarkan hasil pengujian perbandingan algoritma, peningkatan kualitas pengelompokan algoritma *K-Means* dan algoritma *PSO* lebih tinggi 3.28% dibandingkan algoritma *K-Means* yang berdiri sendiri.
3. Penambahan klaster berpengaruh dalam kualitas hasil pengelompokan. Jumlah klaster terbaik berdasarkan pengujian adalah enam klaster dengan nilai *Silhouette Coefficient* tertinggi yaitu 74.01%.

11. DAFTAR PUSTAKA

Ahmadyfard, Alireza dan Modarez, Hamidreza, 2008. *Combining PSO and K-Means to Enhance Data Clustering. IEEE.*

Agusta, Yudi, 2007, *K-Means – Penerapan, Permasalahan, dan Metode Terkait*, Jurnal Sistem dan Informatika. Vol.3.

Anggara, et al., 2016, *Pemilihan Distance Measure Pada K-Means Clustering Untuk Pengelompokan Member Di*

Alvaro Fitness, Jurnal Sistem dan Teknologi Informasi (JUSTIN). Vol. 1, No. 1.

Bopche, Kripa Shankar dan Jain, Anurag, 2016. *A Hybrid Clustering Technique Combining A PSO Algorithm K-Means*. International Journal of Computer Applications. Vol. 137, No. 1.

Cui, Xiaohui, et al., 2005, *Document Clustering Using Particle Swarm Optimization*, *IEEE.*

Dong, Jinxin dan Qi, Minyong, 2009. *A New Algorithm for Clustering Based On Particle Swarm Optimization and K-Means. IEEE.*

Eberhart, RC, Kenedy, J, dan Y, Shi, 2001, *Swarm Intelligent*, New York : Morgan Kaufmann Engelbrecht, AP dan Merwe, DW van der, 2003. *Data Clustering Using Particle Swarm Optimization. IEEE.*

Fayyad, et al., 1996, *From Data Mining To Knowledge Discovery In Databases*, *AAAI*. Vol.17. No.3.

Herdiyeni, Yeni, Bisilisin, Franki Yusuf dan Silalahi, Bib Paruhum, 2014. *Optimasi K-Means Clustering Menggunakan Particle Swarm Optimization pada*

Sistem Identifikasi Tumbuhan Obat Berbasis Citra. Jurnal Ilmu Komputer Agri-Informatika. Vol. 3, No. 1.

Jaganathan, P dan Jaiganesh, S, 2013. *An Improved K-Means Algorithm Combined With Particle Swarm Optimization Approach For Efficient Web Document Clustering. IEEE.*

Kenedy, dan Eberhart, 1995, *Particle Swarm Optimization, IEEE.*

Khoshdel, Hamed dan Saman, Barat, 2012. *A New Hybrid Learning-Based Algorithm for Data Clustering. IEEE.*

Kuo, et al., 2011, *A PSO-Based Intelligent Service Dispatching Mechanism For Customer Expectation Management*, *ELSEVIER.*

Kusuma, Maulian Eka, 2016. *Hybrid Particle Swarm Optimization dan K-Means Untuk Clustering Data Penentuan UKT Proporsional*. Skripsi. FILKOM,

- Universitas Brawijaya.
- Luthfi, dan Kusriani, 2009, *Algoritma Data Mining*, Yogyakarta : Andi Offset, ISBN:9789792908091.
- Macedo, Mariana, Santos, Pedro, Figueiredo, Elliackin, Jr., Clodomir J. Santana, 2017. *Application of PSO-Based Clustering Algorithms on Educational Databases*. IEEE.
- Padma, M.Priyadharshini dan Komorasamy, Mr.G., 2012. *A Modified Algorithm For Clustering Based On Particle Swarm Optimization And K-Means*. IEEE.
- Parsopoulos, Konstantinos E. dan Vrahatis, Michael N., 2010. *Particle Swarm Optimization and Intelligence: Advances and Application*. New York : IGI Global. ISBN: 1615206671, 9781615206674.
- Rivai, dan Sagala, 2014, *Manajemen Sumber Daya Manusia Untuk Perusahaan: Dari Teori ke Praktik (Edisi 3)*, Jakarta : Rajawali Pers, ISBN: 9789797697761
- Shi, Yuhui dan Eberhart, Russell, 1998. *A Modified Particle Swarm Optimizer*. IEEE.
- Shiau, Der-Fang, 2010, *A Hybrid Particle Swarm Optimization For A University Course Scheduling Problem With Flexible Preferences*, ELSEVIER.
- Silpa, B.Venkata, 2015, *An Optimal Power Flow To Improve Power System Security by Using Particle Swarm Optimization*, ELELIJ. Vol. 4, No.2.
- Tan, et al., 2006, *Introduction To Data Mining*, US : Pearson Addison-Wesley, ISBN : 0321420527, 97800321420527.
- Toreini, Ehsan dan Mehrnejad, Maryam, 2011. *Clustering Data with Particle Swarm Optimization Using a New Fitness*. IEEE.
- Wahyuni, Ida, Auliya, Yudha Alif, Rahmi, Asyrofa dan Mahmudy, Wayan Firdaus, 2016. *Clustering Nasabah Bank Berdasarkan Tingkat Likuiditas Menggunakan Hybrid Particle Swarm Optimization dengan K-Means*. Malang: Jurnal Ilmiah Teknologi dan Informasi ASIA. Vol. 10, No. 2. ISSN:0852-730X.
- Wati, 2011, *Sistem Kendali Cerdas : Fuzzy Logic Controller, Jaringan Syaraf Tiruan, Algoritma Genetik, dan Particle Swarm Optimization*, Yogyakarta : Graha Ilmu, ISBN: 9789797567217.
- Wu, Junjie, 2012. *Advances in K-Means Clustering: A Data Mining Thinking*. Germany, Berlin : Springer-Verlag Berlin dan Heidelberg GmbH & Co. KG. ISBN: 3642298079, 9783642298073.
- Yazdani, Danial, Golyari, Sara, dan Meybodi, Mohammad Reza, 2010. *A New Hybrid Approach for Data Clustering*. IEEE.
- Zaki, et al., 2014, *Data Mining And Analysis: Fundamental Concepts And Algorithm*, New York : Cambridge University Press, ISBN:9870521766333