

Integrasi *Particle Swam Optimization* Menggunakan *K-Means* untuk Klasterisasi Pengangguran di Kota Pagar Alam

Deka Agustina¹, Yogi Isro' Mukti², Siti Muntari³

^{1,2,3}Teknik Informatika, Teknik, Institut Teknologi Pagar Alam

¹dekaagustina78@gmail.com ¹yogie.isro.mukti@gmail.com ³muntariaza@gmail.com

Abstract

This research is motivated by the problem of unemployment which is still significant in the city of Pagar Alam which will have an impact on economic and social conditions, the higher the unemployment rate indicates that there is still an imbalance between job seekers and job opportunities themselves. The purpose of this study is to cluster unemployment in the city of Pagar Alam using k-means which is optimized using particle swarm optimization (PSO). The data mining method used is Cross Industry Standardized Process For Data Mining (CRIP-DM) which consists of 6 stages starting from business understanding, data understanding, data preparation, modelling, evaluation and deployment. The attributed used were, year, age, gender, education, and district. Produced 2 cluster, namely cluster_0 as many as 616 people and namely cluster_1 as many as 69 people. The testing phase used the Davies Bouldin index to find out whether the performance of a cluster produced is 0.60, while the test using the Davies Bouldin index which the optimized with particle swarm optimization gets a lower value of 0.59 which means it has succeeded in increasing the cluster result on k-means.

Keywords: CRIP-DM, Davies Bouldin index, k-means, unemployment, Davies Bouldin index.

Abstrak

Penelitian ini dilatar belakangi oleh masalah pengangguran yang masih signifikan di kota Pagar Alam yang akan berdampak pada kondisi ekonomi maupun sosial, semakin tinggi tingkat pengangguran menandakan masih adanya ketimpangan antara pencari kerja dan kesempatan kerja itu sendiri. Tujuan dari penelitian ini adalah untuk mengklasterisasi pengangguran di kota Pagar Alam menggunakan *K-Means* yang dioptimasi menggunakan *Particle Swam Optimization* (PSO). Metode data mining yang di gunakan yaitu *Cross Industry Standard Process For Data Mining* (CRIP-DM) yang terdiri dari 6 tahapan dimulai dari *business understanding*, *data preparation*, *modeling*, *evaluation* dan *deployment*. Atribut yang digunakan yaitu, tahun, umur, jenis kelamin, pendidikan dan kecamatan. Menghasilkan 2 cluster, *cluster_0* sebanyak 616 dan *cluster_1* sebanyak 69. Tahap pengujian menggunakan *Davies Bouldin Index* untuk mengetahui baik tidaknya performa suatu cluster yang di didapatkan nilai 0.60, sedangkan pengujian menggunakan *Davies Bouldin Index* yang sudah dioptimasi dengan *particle swam optimization* mendapatkan nilai lebih rendah yaitu 0.59 berhasil meningkatkan hasil cluster pada k-means.

Kata kunci: CRIP-DM, Davies Bouldin index, k-means, pengangguran, particle swam optimization.

© 2023 Jurnal Pustaka AI

1. Pendahuluan

Pengangguran merupakan masalah yang kerap kali dialami oleh negara berkembang salah satunya

Indonesia. Pengangguran ialah orang yang tidak memiliki pekerjaan tetap atau tidak sama sekali bekerja. Sedangkan pengangguran terbuka (*open unemployment*) dapat didefinisikan sebagai seorang

yang tidak bekerja tetapi sedang dalam tahap pencari pekerjaan dan mempersiapkan usaha untuk mendapatkan pekerjaan. Terjadinya pengangguran dikarenakan banyaknya jumlah tenaga kerja yang tidak seimbang dengan laju pertumbuhan kesempatan kerja, diakibatkan kecilnya lapangan pekerjaan yang ada di suatu wilayah. [1].

Adapun faktor-faktor yang memicu terjadinya tingkat pengangguran diantaranya karena tidak seimbangnya jumlah pencari kerja dengan kesempatan kerja sehingga tidak terwujudnya kesempatan kerja yang setara.[2] Pertumbuhan penduduk berdampak pada bertambahnya jumlah pencari kerja, yang akan mengakibatkan pengangguran, jika hal tersebut tidak diimbangi dengan ketersediaan lapangan pekerjaan [3]. Begitupun tingkat pendidikan dan penduduk usia kerja yang tidak dapat mengikuti kemajuan teknologi juga berpengaruh terhadap meningkatnya angka pengangguran.

Berdasarkan hasil wawancara yang dilakukan di Dinas Sosial Pagar Alam, permasalahan pengangguran di kota Pagar Alam masih cukup signifikan yang akan berpengaruh pada kondisi ekonomi maupun sosial. Dilihat dari korelasi pengaruh sosial akibat dari pengangguran antaranya kan berdampak pada masalah ekonomi dan kemiskinan serta menurunnya tingkat kesejahteraan masyarakat, serta tidak maksimalnya perputaran ekonomi baik secara Mikro maupun Makro dikarenakan terhambatnya transaksi jual beli dan perputaran ekonomi sebagai akibat dari tingkat pengangguran.

Menurut [4] Algoritma *K-Means* merupakan metode data mining Analisis cluster non hirarki mengelompokkan data ke dalam satu atau beberapa *cluster*, sehingga data yang mempunyai karakteristik yang sama akan dikelompokkan ke dalam satu *cluster*, begitupun sebaliknya jika data memiliki karakteristik yang berbeda akan di kelompokkan ke dalam *cluster* yang lain.

Algoritma *K-Means* sangat efisien untuk pemrosesan data dalam jumlah besar, merupakan algoritma yang sederhana untuk mengelompokkan data dan mempunyai kemampuan dalam pencarian lokal yang optimal, namun di samping itu masih lemah dalam sisi pencarian global yang optimal. Maka dari itu diperlukan metode untuk mengoptimalkan algoritma *K-Means*. PSO sendiri terbukti merupakan metode yang baik untuk mengatasi masalah optimasi. [5] PSO sendiri merupakan algoritma yang dapat mengoptimasi pusat *cluster* pada *K-Means* [6].

Dari permasalahan di atas dapat disimpulkan pengangguran dapat menyebabkan berbagai macam faktor permasalahan. Angka pengangguran di kota Pagar Alam sendiri setiap tahunnya tidak tentu dilihat dari grafik di atas. Berbagai macam upaya pemerintah untuk menganggulangi masalah pengangguran masih dilakukan seperti pengadaan pelatihan tenaga kerja,

namun terkadang tidak tepat sasaran dikarenakan masalah tenaga kerja yang tidak kompeten ataupun tidak sesuai keahlian. Jadi perlu dilakukan analisa untuk mengklasterisasi pengangguran. Salah satu metode pengelompokan *clustering* dalam data mining yaitu *K-Means* dapat digunakan. Untuk nantinya dapat diketahui kategori tingkat pengangguran yang termasuk ke dalam cluster tinggi, atau sedang ataupun cluster rendah. Dari hasil pengelompokan yang diperoleh dapat digunakan pemerintah untuk melakukan penanganan terhadap masalah pengangguran di kota Pagar Alam dengan mengkaji faktor yang mempengaruhi pengangguran seperti faktor tingkat pendidikan.

Namun untuk melakukan pemrosesan data pengangguran dalam jumlah yang besar perlu dilakukan optimasi untuk meningkatkan hasil *cluster*. Untuk meningkatkan hasil optimasi disini penulis menggunakan *Particle Swam optimization*. PSO adalah metode pencarian berbasis populasi yang di inialisasikan dengan populai secara acak, digunakan untuk memecahkan masalah optimasi.(Pamsimas, 2020). PSO di dasari oleh suatu perilaku sosial kawanan burung, ikan atau makhluk hidup yang berkelompok lainnya[7]. Maka penulis disini menggunakan *Particle Swam Optimization* untuk pemilihan atribut untuk mengoptimalkan *K-Means* dalam mengklasterisasi pengangguran yang ada di kota Pagar Alam.

[8] Optimasi menggunakan *Particle Swam Optimization* digunakan untuk meningkatkan hasil cluster pada metode *K-Means* yang dibuktikan dengan pengujian *Davies Bouldin Index*, dari hasil pengujian 5 *cluster* yang sudah dioptimasi hasilnya lebih baik ketika sebelum dioptimasi walaupun algoritmanya lebih lambat. Sehingga dapat disimpulkan penggunaan *Particle Swam Optimization* untuk optimasi *K-Means* terbukti tepat setelah di optimasi dengan *Davies Bouldin index*. Hubungan pada penelitian ini juga menggunakan *Particle Swam optimization* untuk meningkatkan hasil *cluster* dan pada tahapan evaluasi juga menggunakan *Davies Bouldin index*.

Berdasarkan penelitian terdahulu yang dilakukan oleh [5]. Dapat disimpulkan algoritma *K-Means* bekerja mengelompokkan atribut data dan pengaruh hasil akurasi serta waktu pemrosesan ditentukan oleh pemilihan jumlah *cluster*. *K-Means* mampu mengatasi masalah yang ada pada dataset, sehingga pada saat klasifikasi dapat mempengaruhi peningkatan waktu pemrosesan. Setelah dilakukan tahap clustering kemudian di proses melalui seleksi fitur dengan *Particle Swam Optimization*., akurasi akan semakin tinggi jika atributnya yang diperoleh semakin sedikit. Hubungan dengan penelitian ini adalah untuk pemilihan seleksi fitur atribut juga menggunakan *Particle Swam Optimization*.

Berdasarkan latarbelakang permasalahan dan penelitian terdahulu di atas maka penulis mengambil judul “Integrasi *Particle Swam Optimization* Menggunakan *K-Means* untuk Klasterisasi Pengangguran di kota Pagar Alam.”

2. Metode Penelitian

Penelitian ini menggunakan metode *Cross Industry Standered Process For Data-Mining* (CRISP-DM). merupakan model proses penambangan data yang digunakan untuk memecahkan masalah oleh para ahli. Penelitian berpedoman pada enam tahap CRISP-DM, atau siklus hidup pengembangan data mining sebagai framwork dari proyek data mining. Metodologi CRISP-DM dijelaskan dalam model proses hirarki, yang terdiri dari empat tingkat abstraksi (dari umum ke khusus): fase, tugas umum, tugas khusus dan contoh proses. Adapun tahapan pada CRISP-DM antara lain sebagai berikut:

2.1. Pemahaman Bisnis (*Bussines Understanding*)

Pada tahapan ini adalah berpusat mengidentifikasi faktor penting meliputi kriteria keberhasilan, untuk memahami suatu kebutuhan dan tujuan dari penambangan data. Penerapan *K-Means* berhubungan langsung dengan data pengangguran di kota Pagar Alam untuk dilakukan klasterisasi sehingga didapatkan hasil pegklasteran.

2.2. Pemahaman Data (*Data Understanding*)

Pada tahap pemahaman data ini berfokus pada proses pengumpulan data, pengecekan kualitas data, dan eksplorasi data sehingga didapatkan informasi yang belum diketahui ataupun tersembunyi untuk dilakukan analisis sehingga diketahui struktur pada data awal agar diperoleh gambaran pola data awal akan dihasilkan dari data mining. Pada penelitian ini menggunakan data pencari kerja di kota Pagar Alam dari tahun 2020-2022. Yang terdiri dari 5 variabel : tahun, jenis kelamin, umur, pendidikan dan kecamatan.

2.3. Pengolahan Data (*Data preparation*)

Pada tahap ini adalah proses pemilihan dan penyiapan data akhir untuk selanjutnya di proses pada tahap pemodelan (*modeling*) dari data mentah. Pada. Pengolahan data ini mencakup pemilihan tabel, variabel serta atribut-atribut data termasuk pembersihan dan transformasi data yang akan dijadikan masukan dalam tahapan *modeling*. Proses pengolahan data mentah tersebut adalah sebagai berikut:

1. Data *cleaning* bertujuan untuk menyelesaikan data dan membuang data yang yang tidak termasuk dalam kriteria. Pada tahap ini yang di atasi adaah *missing value* dan *noice*, kondisi dimana data yang hilang atau tidak lengkap di dalam dataset. Dalam penelitian ini ada 6 data yang digunakan seperti nama pencari kerja, jenis kelamin, umur dan kecamatan. Dari data

tersebut akan dilakukan *cleaning* atau pembersihan data yang kosong atau tidak tepat dengan cara mengisi secara manual.

2. Data *integration* yaitu tahapan untuk menggabungkan menjadi satu kesatuan data yang besar. Pada tahap ini data per tahun digabungkan menjadi satu dataset atau tabel dalam bentuk *excel*.
3. Data *transformation* digunakan untuk mengubah struktur data, format data atau nilai data sedemikian rupa sehingga menghasilkan dataset yang sesuai dengan proses *mining* atau sesuai dengan algoritma yang dirancang. Pada tahap ini dilakukan perubahan atau pengeditan data yang sudah sesuai dengan struktur, format dan nilai agar bisa dilakukan perhitungan menggunakan algoritma *k-means* dengan *particle swam optimization*.

2.4. Pemodelan (*Modelling*)

Pada tahap ini pemodelan dilakukan untuk mendapatkan nilai *cluster* yang optimal sehingga pemodelan dapat di proses dengan menggunakan metode data mining yang dirancang, menggunakan *particle swam optimizaion* dan *k-means*.

Pada tahap ini akan dilakukan dua kali tahapan pemodelan yang pertama menggunakan algoritma *k-means*, selanjutnya dilakukan tahapan pemodelan *k-means* berbasis *particle swam optimization*. Proses pemodelan tersebut menggunakan bahasa pemrograman *python* yaitu sebagai berikut:

1. *K-Means*

Algoritma *K-Means* merupakan algoritma Klasterisasi yang mengelompokan data berdasarkan titik pusat *cluster (centroid)* terdekat dengan data [9]. Tujuan dari *clustering* ini adalah pengelompokan data ke dalam *cluster* berdasarkan tingkat kemiripan, tingkat kemiripan dihitung berdasarkan jarak antara data. Dengan menggunakan persamaan sebagai berikut:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \dots\dots\dots (1)$$

Keterangan:

$D(x,y)$ = Jarak antar data x ke data y

x_i = Data testing ke –i

y_i = Data training ke –i

2. *Particle Swam Optimization*

Menurut [10] *Particle Swam Optimization* adalah teknik berbasis populasi (sejumlah partikel) dengan mengatur posisi dan kecepatan yang mengacu pada

partikel yang optimum untuk mencapai solusi. PSO digunakan untuk memecahkan masalah optimasi. *Particle Swam Optimization* (PSO) didasarkan pada perilaku sekawan burung atau yang meniru perilaku sosial organisme. Berikut ini merupakan rumus kecepatan dan posisi partikel pada *particle swam optimization* oleh kennedy dan Elberhart.

$$v_i(t) = V_{i1}(t-1) + \varphi c_1(P_i - X_i(t-1)) + \varphi c_2(g - X_i(t-1))$$

$$X_{i(t)} = X_i(t-1) + V_1(t)$$

.....(2)

Keterangan:

$V_i(t-1)$ = Kecepatan partikel ke-i pada iterasi ke-i

P_i = Informasi dari lingkungan partikel terbaik dalam suatu populasi

g = Informasi dari lingkungan partikel dalam satu populasi

$X_i(t-1)$ = Posisi saat partikel ke-i pada iterasi ke-i

φ = Vector acak dengan rentang skor 0,1

C_1 = Koefisien dari pembelajaran kongnitif

C_2 = Koefisien pembelajaran sosial

2.5. Evaluasi (*Evaluation*)

Paada tahapan ini peneliti melakukan evaluasi dan validasi untuk mengetahui baik tidaknya suatu performa *cluster* dengan menggunakan metode *davies bouldin index* untuk menentukan apakah model dapat mencapai tujuan yang ditetapkan pada tahap awal (*Business understanding*). Pada tahapan ini akan dilihat hasil perbandingan performa *cluster* dengan menggunakan *k-means* dan *k-means* PSO.

2.6 Penyebaran (*Deployment*)

Pada tahapan ini akan di peroleh pengetahuan serta informasi dari hasil proses analisa yang sudah dibangun. Selain itu penelitian yang dilakukan akan menghasilkan pola dan informasi yang dihasilkan berdasarkan proses yang telah dilakukan pada target awal pada CRIP-DM.

3. Hasil dan Pembahasan

Dalam penelitian penulis telah berhasil membuat model klasterisasi pengangguran di kota Pagar Alam yang sudah disesuaikan dengan menggunakan metode CRIP-DM yang terdiri beberapa tahap yaitu: *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modeling*, *evaluation* dan *Depoylpmnt*.

Proses hasil tersebut dimuai dari proses proses algoritma *k-means* yang dimulai dari penentuan nilai k , memilih titik *centroid* secara acak, memperbarui titik *centroid* sebelum dilakukan *clustering* menggunakan bahasa pemrograman *python* dengan

google colab. Penentuan jumlah *cluster* ditentukan berdasarkan nilai *davies bouldin index* pada *cluster* paling kecil mendekati nol yaitu didapatkan hasil 2 *cluster*, *cluster_0* berjumlah 616 dan *cluster_1* berjumlah 96. Selanjutnya untuk nilai pengukuran jarak antara titik *cluster* menggunakan *davies bouldin index* yaitu 0.60 sedangkan untuk pengukuran jarak titik *cluster* berbasis *particle swam optimization* didapatkan nilai 0.59 yang berarti berhasil meningkatkan hasil *cluster* pada *k-means*.

3.1. Pengolahan Data (*Data Preparation*)

Pada tahapan pengolahan data, Selanjutnya dataset di olah secara manual untuk di akumulasikan sesuai dengan kebutuhan. Data yang akan digunakan yaitu data pencari kerja dari tahun 2020-2022.

Selanjutnya untuk mempermudah pengolahan data perlu dilakukan transformasi data. Data yang belum dapat diolah yang berbentuk nominal/huruf akan di ubah ke dalam bentuk angka/ numerik yang dapat dilihat pada tabel dibawah ini.

Tabel 1. Kriteria

Kriteria	Subkriteria	Nilai
Tahun	-	-
Umur	-	-
Jenis Kelamin	Laki-Laki	1
	Perempuan	2
Pendidikan	SD	1
	SMP	2
	SLTA	3
	Diploma I/II/III	4
	S1	5
Kecamatan	Dempo Utara	1
	Dempo Tengah	2
	Dempo Selatan	3
	Pagar Alam Utara	4
	Pagar Alam Selatan	5

Setelah data sudah diolah menjadi bentuk angka dan disesuaikan dengan kebutuhan, maka akan didapatkan hasil akhir berupa data yang sudah siap untuk dilakukan pada proses tahapan pemodelan. Dapat dilihat pada tabel dibawah ini:

Tabel 2. Transformasi Data

No	Tahun	Umur	JK	Pendidikan	Kecamatan
1	2020	23	1	3	4
2	2020	18	2	3	5
3	2020	21	1	3	5
4	2020	18	2	3	1
5	2020	19	2	3	2
6	2020	23	1	5	5

7	2020	17	1	5	5
8	2020	20	1	3	5
9	2020	18	1	3	5
10	2020	21	1	3	1
11	2020	21	2	3	5
12	2020	19	2	3	4
13	2020	19	1	3	3
14	2020	19	1	3	4
15	2020	23	2	3	2
...
68	2022	22	2	2	5
5					

3.2. Pemodelan (*Modelling*)

Pada tahapan ini akan dilakukan dua kali pemodelan, menggunakan bahasa pemrograman *python*. Pemodelan pertama dilakukan dengan proses pengklasteran dengan menggunakan algoritma *k-means*, kemudian tahapan selanjutnya akan dilakukan pemodelan dengan menggunakan algoritma *k-means* berbasis *particle swarm optimization* untuk mengoptimasi *cluster*.

Tahapan pertama yaitu menginputkan library, pada tahap ini merupakan tahapan utama yang sangat berpengaruh pada hasil program berikutnya. *Library* yang digunakan adalah sebagai berikut:

Program Jurnal

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.cluster import KMeans
```

Library *pandas* digunakan untuk pengolahan *dataframe* dan *numpy* digunakan untuk memproses data dalam bentuk *array*. Sedangkan *library* *matplotlib.pyplot* dan *seaborn* digunakan untuk memvisualisasi data dalam bentuk tabel dan grafik.

Setelah menginput *library* langkah selanjutnya yaitu menginputkan dataset yang akan digunakan, dengan perintah sebagai berikut:

Program Jurnal

```
#Input Data
df = pd.read_excel('datapencarikerja1.xlsx')
```

Selaanjutnya membagi data dengan menggunakan *Cross Validation*. Pengujian data digunakan untuk membagi data secara acak berdasarkan nilai *k-fold* yang ditentukan. untuk menilai kinerja proses dari algoritma. Untuk menguji data menggunakan perintah sebagai berikut:

Program Jurnal

```
# Membagi sampel data secara acak menggunakan KFold
Cross Validation
```

```
from sklearn.model_selection import KFold
```

```
# Use KFold
kf = KFold(n_splits=5, shuffle=True, random_state=1111)
```

```
# Create splits
splits = kf.split(x)
```

```
# Print the number of indices
for train_index, val_index in splits:
    print("Number of training indices: %s" % len(train_index))
    print("Number of validation indices: %s" % len(val_index))
```

Selanjutnya akan ditampilkan hasil dari pembagian data secara acak berdasarkan 5 *k-fold* didapatkan sebanyak 548 data *training* dan 137 *validation indices*. Atau dalam persentase sebanyak 80% *training* dan 20 % *validation indices*.

```
Number of training indices: 548
Number of validation indices: 137
Number of training indices: 548
Number of validation indices: 137
Number of training indices: 548
Number of validation indices: 137
Number of training indices: 548
Number of validation indices: 137
Number of training indices: 548
Number of validation indices: 137
```

Gambar 1. *Split Data*

Langkah selanjutnya yaitu melihat korelasi keterkaitan pada variabel yang mendekati angka 1.0 merupakan variabel yang paling berpengaruh.

	Tahun	umur	jk	pendidikan	kecamatan
Tahun	1.00	0.18	0.01	-0.20	-0.07
umur	0.18	1.00	0.13	-0.02	0.05
jk	0.01	0.13	1.00	0.04	0.06
pendidikan	-0.20	-0.02	0.04	1.00	0.06
kecamatan	-0.07	0.05	0.06	0.06	1.00

Gambar 2. *Korelasi Variabel X dan*

3.2.1 Pemodelan *K-Means*

Selanjutnya menginputkan pemodelan dengan *k-means*, tahapan pertama yaitu menentukan jumlah *k* pada *k-means clustering*, dengan menggunakan perintah sebagai berikut:

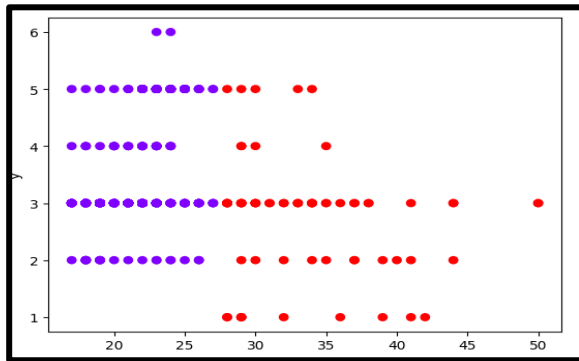
Program Jurnal

```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=2, verbose=0, tol=1e-3, max_iter=300, n_init=20)
```

Setelah menentukan jumlah *cluster*, selanjutnya menentukan titik *centroid* pada masing-masing cluster, dengan menggunakan perintah di bawah ini:

Program Jurnal

```
clus_cent=kmeans.cluster_centers_  
clus_cent
```



Gambar 3. Visualisasi Titik *Centroid*

Selanjutnya untuk menampilkan tabel dengan hasil *cluster*, menggunakan perintah dibawah ini:

Program Jurnal

```
df['Clusters'] = pred  
df
```

	Tahun	umur	jk	pendidikan	kecamatan	Clusters
0	2020	23	1	3	4	0
1	2020	18	2	3	5	0
2	2020	21	1	3	5	0
3	2020	18	2	3	1	0
4	2020	19	2	3	2	0
...
680	2022	19	2	2	2	0
681	2022	28	2	3	5	1
682	2022	40	2	2	5	1
683	2022	27	1	3	5	0
684	2022	22	2	3	5	0

685 rows x 6 columns

Gambar 4. Menampilkan Hasil *Cluster*

Untuk melihat jumlah masing-masing tiap *cluster*, gunakan perintah dibawah ini.

Program Jurnal

```
df['Clusters'].value_counts()
```

```
0    616  
1     69  
Name: Clusters, dtype: int64
```

Gambar 5. Menampilkan Jumlah *cluster*

Selanjutnya mengevaluasi baik tidaknya hasil *cluster* menggunakan *davies bouldin index* dengan menggunakan perintah dibawah ini:

Program Jurnal

```
from sklearn.metrics import davies_bouldin_score  
results = {}  
for i in range(2,6):  
    kmeans = KMeans(n_clusters=i, random_state=30)  
    labels = kmeans.fit_predict(df)  
    db_index = davies_bouldin_score(df, labels)  
    results.update({i: db_index})
```

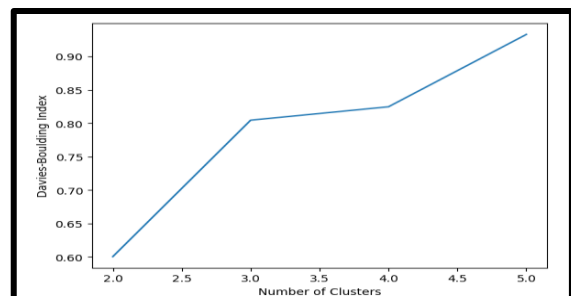
```
{2: 0.6005612925118152,  
3: 0.8046129692181262,  
4: 0.824660990320994,  
5: 0.9328305236512635}
```

Gambar 6. Menampilkan Nilai *Davies Boudin Index*

Dari gambar di atas dari cluster 2-5 nilai DBI yang paling mendekati 0 adalah cluster 2 yaitu didapatkan nilai 0.60 berarti 2 *cluster* yang digunakan sudah tepat. Kemudian untuk memvisualisasi *davies bouldin index* dalam bentuk grafik menggunakan perintah sebagai berikut:

Program Jurnal

```
plt.plot(list(results.keys()), list(results.values()))  
plt.xlabel("Number of Clusters")  
plt.ylabel("Davies-Boulding Index")  
plt.show()
```



Gambar 7. Visualisais *Davies Boudin Index*

Menampilkan hasil *davies bouldin index* dengan menggunakan 2 *cluster* yang sudah ditentukan dari hasil pencarian metode DBI sebelumnya, dengan perintah dibawah ini:

Program Jurnal

```
kmeans = KMeans(n_clusters=2, random_state=30)  
kmeans.fit(df)  
pred = kmeans.predict(df)  
db_index = davies_bouldin_score(df, pred)  
print(db_index)
```


0.6005612925118152

Gambar 8. Hasil DBI 2 Cluster

3.2.2 K-Means Berbasis Particle Swam Optimization

Pada tahap awal pemodelan *k-means* berbasis *particle swam optimization* masih sama dengan *k-means* biasa. Namun perbedaannya adalah sebelum dilakukan pemodelan *k-means* terlebih dahulu ditambahkan *particle swam optimization* sebagai fitur selection. Perlu dilakukan pencarian posisi partikel terbaik an posisi partikel terbaik pada iterasi fungsi.

```
array([[ 23.60808922, -34.64984898],
       [ 9.29568478, 49.55116399],
       [ 32.25471034, -22.71719121],
       [-15.68457796, 46.48579522],
       [-14.38141478, -47.84852285],
       [-37.35822362, 44.36160239],
       [ 22.05879209, -34.26152129],
       [ 4.95263638, -6.87863915],
       [-2.46271972, -7.87195092],
       [-14.94283253, -13.73163099],
       [-45.75662897, -4.01809378],
       [ 22.67513799, 20.36610031],
       [ 21.37688003, 28.74731822],
       [-29.22068239, -45.40915236],
       [-15.51370327, 6.85730159],
       [ 44.74637926, 12.25143105],
       [-43.71861242, 48.09181333],
       [ 20.77351337, 25.65660285],
       [-35.51057813, 29.20514892],
       [-19.82532773, -48.86884456],
       [ 20.78619854, 6.81009764],
       [-25.65873608, -23.86134692],
       [ 12.26018894, -14.11329919],
       [-4.06226203, 45.94269106],
       [ 9.07295016, 46.85177215],
       [ 10.10473335, -35.02028823],
       [ 0.93471382, 27.07411858],
       [ 24.81996128, -36.58399084],
```

Gambar 9. Hasil Pbest

```
The best position is: <FloatingArray>
[-2.6212988335836, 0.6143420772780216]
Length: 2, dtype: Float64 in iteration number 1 The best function is: 8.305523489641594
The best position is: <FloatingArray>
[-2.6250622759650746, 0.6110415391308592]
Length: 2, dtype: Float64 in iteration number 2 The best function is: 8.264323715238348
The best position is: <FloatingArray>
[-2.6250622759650746, 0.6110415391308592]
Length: 2, dtype: Float64 in iteration number 3 The best function is: 8.264323715238348
The best position is: <FloatingArray>
[-2.6250622759650746, 0.6110415391308592]
Length: 2, dtype: Float64 in iteration number 4 The best function is: 8.264323715238348
The best position is: <FloatingArray>
[-2.6250622759650746, 0.6110415391308592]
Length: 2, dtype: Float64 in iteration number 5 The best function is: 8.264323715238348
The best position is: <FloatingArray>
[-2.6250622759650746, 0.6110415391308592]
Length: 2, dtype: Float64 in iteration number 6 The best function is: 8.264323715238348
The best position is: <FloatingArray>
[-2.6250622759650746, 0.6110415391308592]
Length: 2, dtype: Float64 in iteration number 7 The best function is: 8.264323715238348
The best position is: <FloatingArray>
[-2.6250622759650746, 0.6110415391308592]
Length: 2, dtype: Float64 in iteration number 8 The best function is: 8.264323715238348
The best position is: <FloatingArray>
```

Gambar 10. Hasil Pbest Pada Iterasi Function

Selanjutnya melakukan pemodelan *k-means*, pada tahap ini pemodelan dilakukan sama dengan pemodelan *k-means* biasa, pada pemodelan ini akan dilihat perbedaan hasil *davies bouldin index* setelah di optimasi menggunakan *particle swam optimization*.

0.599026754641072

Gambar 11. Hasil DBI setelah dioptimasi dengan PSO

3.3. Evaluasi (Evaluation)

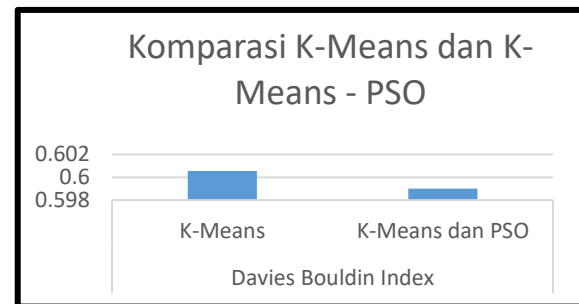
Pada tahapan pengolahan data, Selanjutnya dataset di Setelah dilakukan tahapan pemodelan, selanjutnya dilakukan tahapan evaluasi untuk membandingkan hasil *davies bouldin index* dengan pemodelan *k-means* biasa dan *k-means* berbasis *particle swam optimization*.

Tabel 21. Perbandingan Algoritma

Uji	K-Means	K-Means-PSO
Davies bouldin index	0,60056	0,59902

Berdasarkan hasil tabel perbandingan dari pemodelan *k-means* biasa dengan *k-means* berbasis *particle swam optimization*.

Berdasarkan hasil tabel perbandingan dari pemodelan *k-means* biasa dengan *k-means* berbasis *particle swam optimization*.

Gambar 12. Komparasi *k-means* dan *k-means* – PSO

Berdasarkan gambar perbandingan tersebut dapat dilihat angka DBI dengan menggunakan *k-means* didapatkan nilai 0.60 Sedangkan *k-means* berbasis *particle swam optimization* didapatkan hasil 0.59 yang berarti berhasil meningkatkan hasil optimasi *cluster* karena nilai *cluster* yang didapat lebih mendekati 0 dibandingkan dengan *k-means* biasa.

3.4 Penyebaran (Deployment)

Setelah dilakukan proses evaluasi, deployment merupakan tahapan terakhir yang nantinya hasil dari analisa model dari algoritma *clustering* bisa diterapkan dalam bentuk sistem. Pada penelitian ini proses *deployment* tidak diterapkan karena pada penelitian ini pada batasan masalah yang dilakukan hanya pada proses analisa saja dan tidak sampai ke proses *deployment*.

4. Kesimpulan

Dalam penelitian integrasi *particle swam optimization* dengan *k-means* menggunakan dataset pencari kerja DPMPTSPK Pagar Alam tahun 2020-2022. Penelitian ini menghasilkan klasterisasi pengangguran di kota Pagar Alam. Didapatkan hasil *cluster* sebanyak 2 *cluster* yaitu *cluster_0* dan *cluster_1*, data yang berada pada *cluster_0* sebanyak 616 dataset dan *cluster_1* sebanyak 69 data. Selanjutnya untuk proses pengujian dilakukan dengan menggunakan *davies bouldin index* menggunakan *k-means* biasa didapatkan nilai 0.60. Sedangkan *k-means* yang sudah di optimasi menggunakan *particle swam optimization* didapatkan hasil 0.59. Dengan demikian penggunaan *particle swam optimization* pada *k-means* berhasil meningkatkan hasil *cluster*.

Daftar Rujukan

- [1] S. Muharni and S. Andriyanto, "Penerapan Metode K-Means Clustering Pada Data Tingkat Pengangguran Terbuka," vol. 22, no. 01, pp. 89–99, 2022.
- [2] G. A. Antika, D. Kurnia, and S. Munawaroh, "Analisis Kritis Terkait Efektivitas Program Pelatihan dan Produktivitas Tenaga Kerja dalam Mengurangi Angka Pengangguran Oleh Dinas Tenaga Kerja Kota Cimahi pada Masa Covid Covid-19 Tahun 2021," vol. 6, no. 1, pp. 42–64, 2022.
- [3] M. R. Oktavia, "Pengaruh Inflasi , Indeks Pembangunan Manusia , Dan Upah," vol. 5, no. 2, pp. 94–102, 2021.
- [4] R. P. Primanda, A. Alwi, D. Mustikasari, P. T. Informatika, F. Teknik, and U. M. Ponorogo, "Data Mining Seleksi Siswa Berprestasi untuk Menentukan Kelas Unggulan Menggunakan Metode K-Means Clustering (Studi Kasus di MTS Darul Fikri)," pp. 88–101, 2021.
- [5] O. Herdiana, S. Maulani, and E. A. Firdaus, "Strategi Pemasaran Produk Industri Kratif Menggunakan Algoritma K-Means Clustering Berbasis Particle Swam Optimization," vol. 15, pp. 1–13, 2021.
- [6] I. Wahyuni, Y. A. Auliya, A. Rahmi, and W. F. Mahmudy, "Clustering Nasabah Bank Berdasarkan Tingkat Likuiditas Menggunakan Hybrid Particle Swarm Optimization dengan K-Means," vol. 10, no. 2, pp. 24–33, 2016.
- [7] G. K. Gandhiadi, L. Putu, and I. Harini, "implementasi Fuzzy C-Mean dan Algoritma Particle Swam Optimization untuk Clustering Kabupaten/Kota di Indonesia Berdasarkan Indikator Indeks Pembangunan Manusia," vol. 11, no. 3, pp. 191–198, 2022.
- [8] T. M. Dista and F. F. Abdulloh, "Clustering Pengunjung Mall Menggunakan Metode K-Means dan Particle Swarm Optimization," vol. 6, pp. 1339–1348, 2022, doi: 10.30865/mib.v6i3.4172.
- [9] I. N. M. Adiputra, "Clustering Penyakit Dbd Pada Rumah Sakit Dharma Kerti Menggunakan Algoritma K-Means," *Inser. Inf. Syst. Emerg. Technol. J.*, vol. 2, no. 2, p. 99, 2022, doi: 10.23887/insert.v2i2.41673.
- [10] K. F. Irnanda, A. P. Windarto, and I. S. Damanik, "Optimasi Particle Swarm Optimization Pada Peningkatan Prediksi dengan Metode Backpropagation Menggunakan Software RapidMiner," *J. Ris. Komput.*, vol. 9, no. 1, pp. 122–130, 2022, doi: 10.30865/jurikom.v9i1.3836.
