

DOKUMEN HAK CIPTA

ALGORITMA BARU K-ALLY SEBAGAI ALTERNATIF ALGORITMA KLAUSTERING DAN ANALISISNYA

A. RINCIAN MODEL ALGORITMA

Bagian ini merinci model algoritma yang diusulkan sebagai bagian utama dari pengusulan hak cipta. Rincian model dimulai abstrak dan penjelasan pendahuluan sebagai berikut:

Abstrak

Algoritma klasterisasi merupakan salah satu pilihan dalam membangun machine learning, unsupervised dan mampu mengelompokkan data untuk diberi label. Pada makalah ini diusulkan suatu algoritma clustering yang diharapkan dapat menjadi alternatif algoritma selain algoritma klasterisasi lain seperti kmean, dan sebagainya. Filosofi dari algoritma yang kami usulkan adalah "prinsip membangun persekutuan". Dalam imajinasi kami, dibayangkan ada sejumlah entitas yang bersaing untuk mendapatkan sekutu sebanyak mungkin untuk membangun aliansi. Filosofi ini diilhami oleh perang Ukraina-Rusia, di mana kedua negara bersaing untuk menarik banyak negara ke pihak mereka untuk mendukung mereka. Ditemukan lebih dari tiga variasi dalam upaya membangun algoritma ini, tetapi dalam makalah ini kami mengusulkan variasi pertama sebagai bahan eksperimen. Implementasi algoritma dibuat dalam bahasa pemrograman python menggunakan google colab. Link colab ini kami coba sediakan di dalam makalah agar kode program bisa langsung diakses. Untuk demonstrasi penggunaan algoritma, data bawaan dari perpustakaan *sklearn* digunakan.

Kata Kunci

Algoritma klasterisasi, Perang Ukraina-Rusia, *Clustering*, *Similarity*, Sekutu.

1. Pendahuluan

Klasterisasi merupakan salah satu metode analisis data yang sering digunakan sebagai metode data mining, yang tujuannya adalah untuk mengelompokkan data dengan karakteristik yang sama ke dalam '*region*' yang sama dan data dengan karakteristik yang berbeda ke dalam '*region*' yang lain. Klasterisasi adalah proses mengkategorikan sebuah dataset dalam kelompok record yang mirip dan disebut klaster, proses klasterisasi dilakukan dengan menggunakan kriteria

tertentu [1]. Milligan menulis bahwa jumlah artikel yang menggunakan metodologi klasifikasi meningkat dari 25 pada tahun 1964 menjadi 501 pada tahun 1976. Antara tahun 1958 dan 1973, lebih dari 1.600 artikel ilmiah tentang klasifikasi diterbitkan. Pencarian literatur terpisah menunjukkan bahwa pada tahun 1985 saja, 1.658 referensi ditemukan tentang topik tersebut [2]. Ada beberapa pendekatan yang digunakan dalam mengembangkan metode clustering. Dua pendekatan utama adalah klasterisasi dengan pendekatan partisi dan klasterisasi dengan pendekatan hirarki. Klasterisasi dengan pendekatan partisi atau sering disebut *partition-based clustering* mengelompokkan data dengan cara mengurutkan data yang dianalisis ke dalam klaster-klaster yang ada. Klasterisasi dengan pendekatan hierarkis atau sering disebut dengan *hierarchical clustering* mengelompokkan data dengan membuat suatu hierarki dalam bentuk *dendogram* dimana data yang sejenis akan ditempatkan pada hierarki yang berdekatan dan yang tidak berada pada hierarki yang jauh. Namun, [3] menulis lebih detail bahwa proses algoritma klasterisasi yang berbeda secara garis besar dapat diklasifikasikan berikut *Partitioning-based* (K-Modes, PAM, CLARA, CLARANS, FCM), *Hierarchical-based* (BIRCH, CURE, ROCK, CHAMELEON), *Density-based* (DBSCAN), OPTIK, DBCLASD, DENCLUE), *Grid-based* (Wave-Cluster, STING, CLIQUE, OptiGid), *Model-based* (EM, COBWEB, CLASSIT, SOMs).

Ada banyak ide yang telah diajukan untuk memperkaya jumlah variasi metode klastering. Beberapa di antaranya seperti algoritma C2P, algoritma klasterisasi baru untuk basis data spasial besar, yang mengeksplorasi metode akses spasial untuk penentuan pasangan terdekat. [4], juga [5], makalah ini menyajikan algoritma propagasi probabilitas yang ditingkatkan untuk klasterisasi yang *peakclustering-based* kepadatan berdasarkan *natural nearest neighborhood* (DPC-PPNNN). Dengan memperkenalkan gagasan tentang *natural nearest neighborhood* dan propagasi probabilitas, DPC-PPNNN mewujudkan proses klasterisasi nonparametrik dan membuat algoritma dapat diterapkan untuk kumpulan data yang lebih kompleks. Pada percobaan pada beberapa dataset, DPC-PPNNN terbukti mengungguli DPC, K-means dan DBSCAN. Ide lain adalah [6], penelitian ini mengusulkan algoritma klasterisasi semi-otomatis *hyperparameter tuning framework* yang dirancang untuk pengembangan persona, HyPersona, untuk mulai menyelesaikan tantangan dengan penyetelan algoritma klasterisasi untuk pengembangan persona otomatis. Andreas Lang dan Erich Schubert [7] memperkenalkan fitur klaster pengganti pada metode klasterisasi BIRCH yang tidak memiliki masalah numerik, yang tidak jauh lebih mahal untuk dipelihara, dan yang membuat banyak perhitungan lebih sederhana dan, karenanya, lebih efisien.

Ide lain, [8] menggunakan algoritma *step whale optimization* yang dimodifikasi hibrida dengan pencarian tabulasi untuk mendapatkan algoritma klasterisasi data yang lebih baik. Borlea juga menulis makalah [1] yang menyajikan cara meningkatkan klaster yang dihasilkan oleh algoritma K-means dengan memproses klaster yang dihasilkan dengan algoritma pembelajaran yang *supervised*. Pendekatan yang diusulkan difokuskan pada peningkatan kualitas klaster yang dihasilkan dan bukan pada pengurangan waktu pemrosesan. Juga Rehman dan Belhaouari [9] telah mengusulkan algoritma klasterisasi non-parametrik baru yang didasarkan pada konsep bagi-dan-gabung. Dalam hal klasterisasi data yang tidak berlabel, [10], Teng Li et al mengusulkan algoritma *ensemble agglomerative hierarchical clustering* berdasarkan teknik klaster dan *novelty* untuk *similarity measurement*. Berbagai penelitian di beberapa bidang yang juga terkait dengan upaya menemukan metode klasterisasi yang lebih baik, terutama untuk penerapan metode klasterisasi di bidangnya, dapat dilihat pada [11] untuk studi degradasi fuel cell stack, [12] untuk memecahkan masalah global masalah pengoptimalan.

Dalam makalah ini, algoritma klasterisasi baru juga diusulkan. Algoritma ini diharapkan dapat menjadi variasi baru dari klasterisasi yang tergolong berbasis *Partitioning*. Hal ini dikarenakan proses klasterisasi mirip dengan *k-mean*. Algoritma yang diusulkan dalam makalah ini didasarkan pada prinsip "kompetisi untuk sekutu".

1.1. Latar Belakang dan Filosofi

Pada 24 Februari 2022, delapan tahun setelah penggelaran strategi tidak langsung militer dan non-militer terpadu melawan Kiev, Vladimir Putin memutuskan untuk memulai perang terbuka melawan Ukraina [13], [14], dan [15] . Strategi politik Rusia ini memicu perang yang mengakibatkan upaya kedua negara untuk mendapatkan dukungan dari negara lain. Negara-negara pendukung dapat dilihat sebagai membentuk aliansi pendukung atau kelompok negara-negara pendukung. Konteks kompetisi Ukraina-Rusia ini dapat dimodelkan menjadi model kompetisi. Model kompetisi ini menginspirasi lahirnya algoritma klasterisasi yang diusulkan dalam makalah ini.

Algoritma pengelompokan yang diusulkan terutama terinspirasi oleh peristiwa perang Ukraina-Rusia. Sebenarnya ini adalah model persaingan perang untuk mencari sekutu antara dua negara. Pemodelan perang ini menjadi algoritma klasterisasi yang diharapkan mampu

mengelompokkan data menjadi beberapa klaster. Dalam model sebenarnya, ini adalah kompetisi untuk menemukan sekutu yang membentuk dua klaster negara, yaitu klaster negara pendukung Ukraina dan klaster negara pendukung Rusia. Setelah dimodelkan ke dalam algoritma pengelompokan, dua pusat klaster dapat diperluas ke sejumlah N pusat klaster.

1.2. Pembuatan Model

Proses penyusunan algoritma ini dimulai dengan memodelkan kompetisi mencari sekutu antara kedua negara. Proses mencari sekutu dapat dirumuskan dalam beberapa proposisi model sebagai berikut:

Tabel 1. Model untuk membentuk aliansi dukungan

Model: Persaingan untuk mencari sekutu
Input: Rusia dan Ukraina dan semua negara di dunia
<p>Proposisi 1 : Rusia dan Ukraina saling berperang atau berkompetisi mencari dukungan dan keduanya saling berusaha membangun aliansi (klaster persekutuan).</p> <p>Proposisi 2 : Terdapat beberapa kesamaan pandangan politik yang dijadikan dasar untuk menarik negara-negara lain untuk bergabung di dalam aliansi. Kesamaan ini digunakan untuk memetakan seluruh negara yang potensial menjadi aliansi. Ini membentuk ruang potensial untuk beraliansi. Kesamaan-kesamaan ini menjadi atribut-atribut dataset negara-negara dan menjadi sumbu-sumbu ruang untuk membangun space politik. Sebuah negara adalah sebuah titik di dalam space politik tersebut.</p> <p>Proposisi 3 : Rusia dan Ukraina yang berkompetisi menggunakan kesamaan itu untuk merekrut negara-negara lain ke dalam aliansi.</p> <p>Proposisi 4 : Setiap negara yang dibujuk untuk membentuk aliansi memiliki pandangan politik atau kebijakan politik yang menjadi pertimbangan untuk berpihak kepada salah satu negara yang berkompetisi.</p> <p>Proposisi 5 : Setiap negara memutuskan untuk berpihak berdasarkan ukuran besar benefit dan cost politik. Benefit yang maksimum dan cost yang minimum menjadi dasar untuk menentukan pilihan kepada negara mana dia membentuk aliansi.</p> <p>Proposisi 6 : Ukuran untuk menentukan aliansi ini dinamakan sebagai jarak politik. Jarak politik berbanding terbalik dengan benefit politik dan berbanding lurus dengan cost politik. Sehingga semakin kecil jarak politik ke suatu negara maka semakin memperbesar kemungkinan untuk bergabung dengan klaster aliansi negara tersebut.</p> <p>Proposisi 7 : Negara yang sudah bergabung ke dalam aliansi dapat menjadi duta baru untuk mengajak negara lain bergabung ke dalam aliansi.</p>
Output 1: Klaster sekutu Ukraina
Output 2: Klaster sekutu Rusia

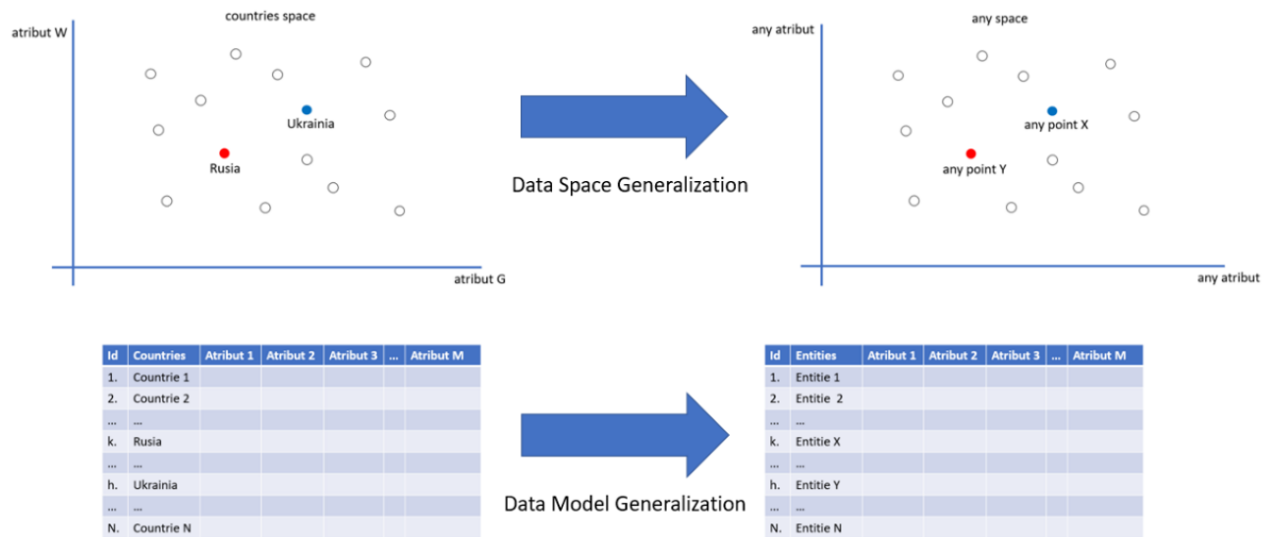
1.3. Interpretasi Model

Model kompetisi mencari sekutu pada tabel 1, dapat ditafsirkan secara lebih rinci agar dapat

diimplementasikan menjadi sebuah algoritma. Penafsiran ini adalah sebagai berikut:

1.3.1. Penjelasan dan penafsiran input, proposisi 1, 2

Semua negara, termasuk Rusia dan Ukraina membangun sebuah model dataset, dimana setiap negara direpresentasikan oleh atribut-atribut. Atribut-atribut boleh apa saja yang dapat membawa benefit atau cost terhadap sebuah negara jika dibandingkan dengan atribut-atribut yang sama dengan negara lainnya. Akan tetapi, jika kita membawa kepada penafsiran universal maka setiap negara hanyalah sembarang titik data atau sembarang rekord. Sehingga sekumpulan negara yang membangun dataset, dapat ditafsirkan sebagai sembarang dataset. Dalam hal ini, penafsiran model adalah generalisasi ruang data dan model data ke sembarang data set.



Gambar 1. Generalization as interpretation of space and model to any dataset

Demikian pula, bahwa Rusia dan Ukraina sebagai pusat aliansi dapat ditafsirkan sebagai sembarang titik data atau sembarang rekord di dalam dataset yang dipilih sebagai titik pusat aliansi. Penafsiran ini dapat diperluas dengan menyatakan bahwa kita dapat memilih k buah sembarang titik data atau k buah titik pusat aliansi untuk k aliansi. Setiap titik pusat aliansi nantinya aktif membangun aliansi dengan titik-titik lain dengan cara mengukur jarak_politik antara titik lain dengan titik pusat aliansi tersebut.

Setiap negara memiliki kesamaan untuk dapat menjadi patokan untuk bergabung ke dalam aliansi. Kesamaan itu adalah kesamaan atribut yang sama untuk semua titik data, dan setiap atribut bisa digunakan untuk menghitung jarak_politik dengan negara atau titik lain. Misal negara X

memiliki atribut W maka sembarang negara juga memiliki atribut W, yang artinya bahwa setiap titik data memiliki atribut W. Ini membangun model data untuk dataset.

1.3.2. Penjelasan dan penafsiran proposisi 3, 4, 5 dan 6

Setiap negara mengukur jarak_politik dirinya dengan setiap negara pusat aliansi (Rusia dan Ukraina), misal jika negara X memiliki jarak politik ke Ukraina lebih kecil dibanding ke Rusia, maka negara itu akan memilih untuk beraliansi dengan Ukraina. Jarak_politik di dalam penafsiran ini adalah sebuah ukuran yang menentukan apakah sebuah negara berpihak kepada aliansi tertentu. Jarak_politik sebagai penentu ditafsirkan sebagai memiliki sifat bahwa semakin kecil jarak politiknya maka semakin cenderung dia beraliansi kepada pihak yang terdekat. Karena itu jarak_politik ditafsirkan sebagai berikut:

“Jarak_politik berbanding terbalik dengan benefit hubungan antar dua negara dan berbanding lurus dengan cost hubungan antar dua negara”

Dengan demikian, semakin besar benefit hubungan kedua negara maka semakin kecil jarak_politiknya yang berarti semakin besar kemungkinan kedua negara membentuk aliansi demikian sebaliknya untuk nilai cost hubungan kedua negara. Berdasarkan penafsiran ini maka jarak_politik didefinisikan sebagai:

$$\text{Jarak_politik dua negara} = \begin{cases} |\text{jarak antar atribut cost}| / |\text{jarak antar atribut benefit}| & \text{jika atribut benefit ada} \\ 1 / |\text{jarak antar atribut benefit}| & \text{jika atribut cost tidak ada} \\ |\text{jarak antar atribut cost}| & \text{jika atribut benefit tidak ada} \end{cases}$$

1.3.3. Penafsiran proposisi 7

Di dalam model ini, setiap negara yang baru menjadi anggota aliansi dapat mengajak negara lain untuk masuk menjadi anggota aliansi. Secara umum, secara paralel, setiap anggota yang lain dapat juga merekrut anggota lain untuk masuk ke dalam aliansi.

1.4. Implementasi Model Ke Dalam Algoritma Klasterisasi

Model yang telah dibuat pada tabel 1 di atas, dapat diimplementasikan ke dalam berbagai variasi algoritma untuk melakukan klasterisasi. Variasi-variasi itu untuk sementara dapat ditunjukkan sebagai berikut.

Dari sisi siapa yang akan melakukan perekrutan untuk masuk ke dalam aliansi:

- (1) Klasterisasi dimana hanya titik data (negara) yang baru bergabung ke dalam aliansi yang dapat merekrut titik data lain (negara lain) untuk masuk ke dalam aliansi. Ini berarti, proses perekrutan titik data untuk masuk ke dalam klaster (alianasi) adalah terjadi secara linier, merambat dari satu teman ke teman yang lain.
- (2) Klasterisasi dimana semua titik data yang bergabung di dalam aliansi adalah dapat aktif secara bersamaan melakukan perekrutan untuk merekrut negara-negara lain. Dalam hal ini, proses klasterisasi atau perekrutan untuk masuk ke dalam klaster adalah terjadi secara paralel.

Dari sisi perumusan jarak politik, ada beberapa variasi di dalam merumuskan jarak politik untuk menentukan apakah sebuah titik data termasuk ke dalam klaster tertentu (alianasi) atau tidak. Di dalam makalah ini, untuk mendemonstrasikan implementasi algoritma dari model, perumusan jarak politik dibuat dengan melakukan penyederhanaan. Secara umum penyederhanaan dilakukan sebagai berikut:

Tabel 2. Himpunan assumi penyederhanaan

Himpunan assumi penyederhanaan untuk implementasi model ke algoritma	
Assumsi 1	: Semua atribut adalah atribut cost
Assumsi 2	: Jarak_politik yang digunakan adalah jarak euclid antar atribut.
Assumsi 3	: Proses perekrutan titik data untuk masuk ke dalam klaster (alianasi) dilakukan oleh titik data yang baru saja direkrut masuk ke dalam klaster terkait. Ini artinya bahwa berlaku prinsip “teman merekrut teman” merambat secara linier.
Assumsi 4	: Untuk iterasi pertama pembentukan klaster (alianasi), titik pusat klaster adalah ditentukan secara acak. Misal ditentukan sebanyak k titik pusat klaster, maka k titik data diambil secara acak dari dataset yang ada.
Assumsi 5	: Untuk iterasi berikut, titik pusat klaster adalah centroid dari setiap klaster yang terbentuk diiterasi sebelumnya. Kemudian proses perambatan “teman merekrut teman” dimulai pada centroid tersebut sampai iterasi selesai.
Assumsi 6	: Kriteria penghentian iterasi ditentukan oleh jumlah iterasi yang diinginkan atau oleh treshhold yang ditentukan. Treshhold mengukur jarak antara centroid iterasi sebelumnya dengan centroid iterasi berikutnya.

Untuk menyesuaikan dengan filosofi dari algoritma ini ketika dibentuk maka algoritma ini diberi nama k-ally. Di bawah ini adalah perumusan implementasi model tabel 1 ke dalam algoritma setelah dilakukan penyederhanaan sebagai berikut:

Tabel 3. Implementasi model ke dalam algoritma

Algoritma: k-ally 1.0 (Implementasi 1.0 model tabel 1)	
Input: sembarang dataset	
Step	Deskripsi
Step 0	: START

Step 1 : Tambahkan kolom kelas pada dataset.	Menambahkan kolom untuk melabeli klaster
Step 2 : Pilih k buah acak record. Diperoleh sebuah array pusat_sekutu	Pemilihan titik pusat klaster sebanyak k buah secara acak pada dataset. Titik pusat sekutu. Dalam hal ini asumsi 4 pada tabel 2 diterapkan.
Step 3 : Masing kolom kelas dari k buah titik terpilih ditandai dengan k buah label.	Setiap titik pusat sekutu yang terpilih dilabeli sesuai label klaster yang diberikan. Misal $k=2$ maka dibuat C1 dan C2 sebagai label klaster dan disematkan kepada setiap titik pusat sekutu yang sesuai.
Step 4 : Untuk setiap titik x selain k terpilih dan selain x terlabel, ukur jaraknya ke titik pusat_sekutu[i]. $i=1,2,3,\dots,k$	Untuk setiap titik data di dataset (setiap record) ukur jaraknya (jarak_politik) kepada setiap titik pusat sekutu (titik pusat klaster) terpilih. Pengukuran jarak menggunakan jarak euclid dimana asumsi 1 dan 2 pada tabel 2 diterapkan.
Step 5 : Jika x terdekat ke pusat_sekutu[i], maka tandai x dengan label pusat_sekutu[i]. Yaitu $x[label]=pusat_sekutu[i][label]$	Jika x adalah salah satu dari record dataset, terdekat ke titik pusat sekutu ke-i, maka labeli x dengan label klaster dari titik pusat sekutu ke-i tersebut. Pelabelan dilakukan pada kolom label yang ditambahkan pada step 1 sebelumnya.
Step 6 : Kemudian pusat_sekutu[i]=x.	Kemudian ganti posisi titik pusat sekutu ke-i dengan titik x. Sekarang x menjadi titik pusat sekutu baru dan bertanggung jawab berikutnya untuk merekrut titik lain menjadi anggota aliansi (klaster). Dalam hal ini, filosofi "teman merekrut teman" atau asumsi 3 pada tabel 2 diterapkan.
Step 7 : Kembali ke Step 4.	Proses perekrutan anggota aliansi berlangsung mulai dari step 4 sampai step 6. Proses ini berlangsung secara berulang (looping) sampai semua titik x (semua record) pada dataset selesai direkrut. Looping dari step 4 sampai step 7.
Step 8 : Klasterisasi berdasarkan ally selesai.	Proses perekrutan anggota aliansi untuk iterasi pertama (iterasi 1)
Step 9 : Hitung centroid semua kelas.	Hitung centroid seluruh aliansi (klaster) yang terbentuk.
Step 10 : Revisi k acak menjadi: pusat_sekutu[i]=mean{x x bersekutu dengan pusat_sekutu[i]}.	Revisi seluruh titik pusat sekutu yang telah terjadi di iterasi 1 dengan mengganti seluruh titik pusat sekutu dengan centroid aliansi. Dalam hal ini, asumsi 5 pada tabel 2 diterapkan
Step 11 : Hitung Jarak rata-rata (JR) semua pusat_sekutu[i] di iterasi	Hitung jarak rata-rata atau Mean Square Error (MSE) antara titik pusat sekutu iterasi sebelumnya dengan titik pusat sekutu yang baru.

baru ke pusat_sekutu[i] iterasi sebelumnya.	
Step 12 : Periksa apakah jumlah iterasi sudah terpenuhi atau JR \leq treshold?	Assumsi 6 dari tabel 2 diterapkan.
Step 13 : Jika Ya maka algoritma SELESAI.	Assumsi 6 dari tabel 2 diterapkan.
Step 14 : Jika belum maka ulang langkah 4.	Assumsi 6 dari tabel 2 diterapkan.
Step 15 : SELESAI.	
Output 1: Klaster data sebanyak $h \leq k$ klaster	
Output 2: h buah centroid	

2. Method

Penelitian untuk menghasilkan algoritma baru ini berdasarkan tahapan metode yang dilakui sebagai berikut:

- (3) Studi pustaka, dengan meninjau filosofi dan review konsep-konsep algoitma klasterisasi sebelumnya sepanjang tahun 2022 dan sebelumnya. Juga meninjau penerapan algoritma-algoritma terebut di berbagai bidang. Juga menelaah fenomena dan konteks perang Rusia-Ukrainia untuk merumuskan suatu model yang sederhana dari pada pembentukan dukungan antar negara atau aliansi di dalam fenomena perang ini, dan diharapkanj dapat digunakan untuk membangun sebuah algoritma klasterisasi yang berdasarkan kecendrungan untuk membangun aliansi.
- (4) Perumusan model pembentukan aliansi.
- (5) Penyederhanaan model untuk memperoleh berbagai variasi penerapan model menjadi algoritma pembentukan aliansi.
- (6) Memilih salah satu himpunan penyederhanan untuk digunakan membangun algoritma pembentukan aliansi.
- (7) Membuat pseudocode algoritma pembentukan aliansi.
- (8) Memberi nama algoritma pembentukan aliansi sebagai k-ally berdasarkan filsosfi pembentukannya dan memaksudkannya sebagai salah satu algoritma baru klasterisasi.
- (9) Implementasi algoritma k-ally ke dalam kode dalam bahasa pemrograman python.
- (10) Testing kode algoritma
- (11) Penjelasan hasil-hasil klasterisasi kode algoritma.

3. Hasil dan Pembahasan

Untuk menjadi bahan diskusi bagi algoritma yang dihasilkan di dalam penelitian ini, hendak dibicarakan analisis tentang kompleksitas komputasi, implementasinya ke dalam koding dan diskusi tentangnya.

3.1. Analisis dan diskusi tentang kompleksitas algoritma

Untuk inisiasi jumlah k klaster yang tidak terlalu banyak, perhitungan kompleksitas dapat didekati sebagaimana dekripsikan di tabel 4

Tabel 4. Perhitungan kompleksitas algoritma

Kompleksitas algoritma k-ally		
Input: dataset berukuran n buah rekord		
Range step	Kompleksitas	Deskripsi
Step 1 – step 3	$O(n)$	$O(\text{Step 1}) + O(\text{step 2}) + O(\text{step 3}) = O(1) + O(k) + O(n)$ $k \leq n$ untuk n adalah ukuran dataset
Step 4 – step 7	$O(n^2)$	$O(\text{step 4 ke step 7}) = O(n)$ Tetapi $O(\text{looping sebanyak } n \text{ kali dari step 4 ke step 7}) = O(n^2)$
Step 8 – step 13	$O(n)$	Dihitung berdasarkan bahwa tidak ada looping, jumlah step diambil n sebagai standar.
Step 14 – step 4	$O(n^3)$	Ini karena looping step 14 – step 4 memuat juga looping step 4 – step 7 sehingga looping $O(n^2)$ dikalikan dengan jumlah putaran iterasi. Jumlah putaran iterasi diambil n sebagai standar. Sehingga diperoleh $n.O(n^2)$ yaitu diambil $O(n^3)$
Total kompleksitas	$O(n) + O(n^2) + O(n) + O(n^3) = O(n^3)$	Karena $O(n) < O(n^2) < O(n^3)$ sehingga bisa didekati dengan $O(n^3)$

Perhitungan kompleksitas pada tabel 4 masih berupa perkiraan kasar dengan tidak menghitung secara rinci pada subprosedur atau subfungsi seperti fungsi penghitung jarak dan fungsi penghitung MSE dan dengan perkiraan penentuan k yang relatif kecil dibanding ukuran dataset.

3.2. Diskusi tentang implementasi kode

Untuk implementasi algoritma ke dalam kode python. Implementasi kode tersebut disajikan pada tabel 5. Kekurangan implementasi ini adalah belum menerapkan step 12 dan step 13 dari algoritma, sehingga threshold belum menjadi syarat penghentian iterasi. Akan tetapi, diharapkan tabel 5 sudah cukup menyajikan implementasi kode untuk dapat melihat bagaimana algoritma bekerja melakukan klasterisasi.

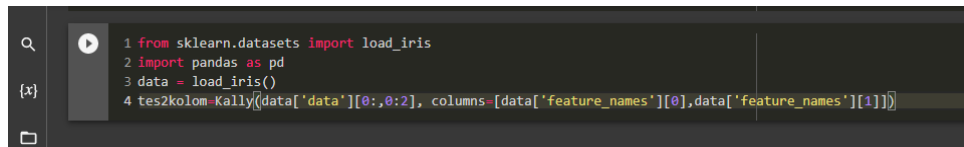
Tabel 5. Implementasi kode algoritma k-ally.1.0

Kode: Algoritma k-ally 1.0	
Language: Python	
Complete source code: https://github.com/elangbijak4/k-ally.1.0	
Library: numpy, panda, random, math	
Class name: Kally	
Property: data, columns, data frame, iteration, treshhold, mean, label.	
<p>Method pendukung:</p> <pre>label(self,x),copy(self,x),pltcolor(self,lst),plot(self,lst,x,y,s=50), distance_ally(self,p,q),mse(self,vektor1,vektor2),build_centroid_random(self,k,min,max),calculate_mean(self,data_frame,pusat_sekutu,komponen)</pre> <p>#Method untuk pelabelan klaster,copy variabel ke alamat memory yang berbeda, pemetaan warna titik pada plotting, fungsi jarak, MSE(Mean Squarred Error), fungsi random, menghitung mean setiap klaster.</p>	
Detail Kode	Deskripsi
Method utama: <code>__init__(self,data,columns,k=3,min_index_data=0,iteration=10,treshhold=0.05) -> None</code>	Konstruktor
<code>for i in self.data_frame:</code> <code>self.data_frame['label']=None</code>	Step 1 dari algoritma pada tabel 3
<code>self.pusat_sekutu=self.build_centroid_random(k,min_index_data,max(self.data_frame.index))</code>	Step 2 dari algoritma pada tabel 3
<code>self.label(self.pusat_sekutu)</code> <code>p=self.copy(self.pusat_sekutu)</code>	Step 3 dari algoritma pada tabel 3
<code>for t in range(self.iteration):</code>	Step 14 dari algoritma pada tabel 3
<code>→for x in self.data_frame.index:</code>	Step 4 dari algoritma pada tabel 3
<code>→if self.data_frame.loc[x]['label'] not in self.label:</code>	
<code>→jarak_list=[]</code>	
<code>→q=[]</code>	
<code>→for i in self.komponen:</code>	
<code>→q=q+[self.data_frame.loc[x][i]]</code>	
<code>→for y in p:</code>	
<code>→jarak_list=jarak_list+[self.distance_ally(y,q)]</code>	
<code>→for w in range(len(jarak_list)):</code>	Step 5, step 6 dan step 7 dari algoritma pada tabel 3
<code>→if jarak_list[w]==min(jarak_list):</code>	
<code>→self.data_frame.at[x,'label']=C[w]</code>	
<code>→self.centroid[w]=x</code>	
<code>→p[w]=[self.data_frame.at[x,self.data_frame.columns[i]]for i in self.komponen]</code>	Step 8 dan step 9 dari algoritma pada tabel 3
<code>→self.mean=self.calculate_mean(self.data_frame,self.pusat_sekutu,self.komponen)</code>	
<code>→rerata_mse=0</code>	Step 11 dari algoritma pada tabel 3

→for i in range(len(p)):	
→rerata_mse=rerata_mse+self.mse(p[i],self.mean[i])	
→p=[i for i in self.mean]	Step 10 dari algoritma pada tabel 3
→if self.label==C :	
→self.label=K	
→else:	
→self.label=C	
→lst=[i for i in self.data_frame[self.data_frame.columns[0:len(self.komponen)]]]	
→cols=self.pltcolor(lst)	Plotting
→self.data_frame.plot.scatter(x=self.data_frame.columns[0],y=self.data_frame.columns[1],s=50,c=cols)	
Output: Dataset dengan label klaster, mse, plotting	

3.3. Testing kode algoritma

Pengujian kode algoritma dilakukan pada jupyter notebook colab google. Data yang digunakan adalah dataset load_iris pada sklearn.datasets, dengan memilih jumlah klaster awal k=3, jumlah iterasi dibatasi 10, dan hanya dua atribut pada dataset yang digunakan, yaitu atribut “sepal length” dan atribut “sepal width” agar plotting 2D dapat dilakukan. Berikut ini *screenshoot* pengujian algoritma. Gambar 2 menyajikan *screenshoot* kode saat dijalankan.



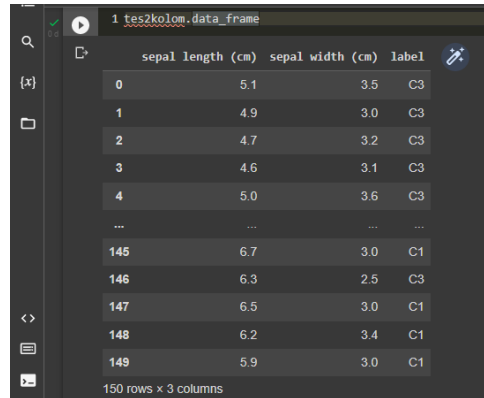
```

1 from sklearn.datasets import load_iris
2 import pandas as pd
3 data = load_iris()
4 tes2kolom_Kally([data['data'][0:0:2], columns=[data['feature_names'][0],data['feature_names'][1]]])

```

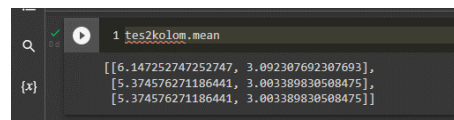
Gambar 2. Menjalankan kode algoritma pada colab google

Hasil eksekusi kode pada gambar 2, dapat dilihat pada gambar 3 yang menyajikan hasil pelabelan klaster dari dataset, gambar 4 menyajikan centorid hasil akhir pada iterasi ke-10 dan gambar 5 menyajikan hasil plotting klaster.



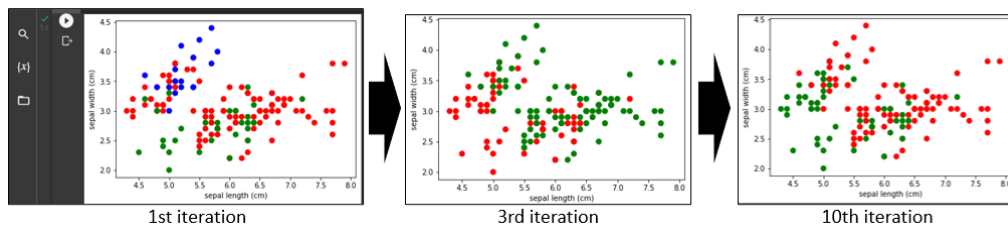
	sepal length (cm)	sepal width (cm)	label
0	5.1	3.5	C3
1	4.9	3.0	C3
2	4.7	3.2	C3
3	4.6	3.1	C3
4	5.0	3.6	C3
...
145	6.7	3.0	C1
146	6.3	2.5	C3
147	6.5	3.0	C1
148	6.2	3.4	C1
149	5.9	3.0	C1

Gambar 3. Hasil klasterisasi pada dataset load_iris, C1 dan C3 adalah klaster iterasi ke-10



```
[[6.147252747252747, 3.002307692307693],
 [5.374576271186441, 3.003389830508475],
 [5.374576271186441, 3.003389830508475]]
```

Gambar 4. Centroid terakhir setelah 10 iterasi



Gambar 5. Plot klasterisasi pada iterasi ke-1, ke-3 dan ke-10

4. Kesimpulan

Algoritma yang pada mulanya berdasarkan filosofi kompetisi pembentukan aliansi diantara dua negara yang berperang, Rusia-Ukraina, telah berhasil dibuat dan demonstrasikan seperti pada gambar 2, 3, 4 dan 5. Walau demikian algoritma ini masih memiliki beberapa kekurangan, terutama karena diimplementasikan dengan beberapa batasan pada tabel 2. Batasan-batasan yang dikemukakan untuk menyederhanakan pembuatan kode algoritma untuk lebih cepat memperoleh hasil demonstrasi dari model yang dibuat. Percobaan menjalankan kode algoritma hanya menggunakan 2 sumbu atau atribut pada dataset, demikian untuk mempermudah melihat plot data hasil klasterisasi, akan tetapi kemampuan kode algoritma adalah dapat melakukan klasterisasi dengan sembarang jumlah atribut dan sembarang jumlah klaster k yang ditetapkan.

5. Acknowledgement

Penulis mengucapkan terima kasih kepada Universitas Muhammadiyah Ponorogo sebagai

pihak yang telah mendukung mendanai keberlangsungan penelitian ini dan kepada LPPM Universitas Muhammadiyah Ponorogo sebagai pihak yang telah mendukung dan mensupervisi penelitian ini.

6. References

- [1] I. D. Borlea, R. E. Precup, and A. B. Borlea, "Improvement of K-means Cluster Quality by Post Processing Resulted Clusters," in *Procedia Computer Science*, 2021, vol. 199, pp. 63–70. doi: 10.1016/j.procs.2022.01.009.
- [2] G. W. Milligan and M. C. Cooper, "Methodology Review: Clustering Methods," *APPLIED PSYCHOLOGICAL MEASUREMENT*, vol. 11, no. 4, pp. 329–354, 1987, [Online]. Available: <http://www.copyright.com/>
- [3] A. Fahad *et al.*, "A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis," *IEEE Trans Emerg Top Comput*, vol. 2, no. 3, pp. 267–279, 2014.
- [4] A. Nanopoulos, Y. Theodoridis, and Y. Manolopoulos, "C 2 P: Clustering based on Closest Pairs," *Proceedings of The VDLB Conference, Roma, Italy, 2001*, 2001.
- [5] W. Zuo and X. Hou, "An improved probability propagation algorithm for density peak clustering based on natural nearest neighborhood," *Array*, vol. 15, p. 100232, Sep. 2022, doi: 10.1016/j.array.2022.100232.
- [6] E. Ditton, A. Swinbourne, and T. Myers, "Selecting a clustering algorithm: A semi-automated hyperparameter tuning framework for effective persona development," *Array*, vol. 14, p. 100186, Jul. 2022, doi: 10.1016/j.array.2022.100186.
- [7] A. Lang and E. Schubert, "BETULA: Fast clustering of large data with improved BIRCH CF-Trees," *Inf Syst*, vol. 108, Sep. 2022, doi: 10.1016/j.is.2021.101918.
- [8] K. K. A. Ghany, A. M. AbdelAziz, T. H. A. Soliman, and A. A. E. M. Sewisy, "A hybrid modified step Whale Optimization Algorithm with Tabu Search for data clustering," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 3, pp. 832–839, Mar. 2022, doi: 10.1016/j.jksuci.2020.01.015.
- [9] A. U. Rehman and S. B. Belhaouari, "Divide well to merge better: A novel clustering algorithm," *Pattern Recognit*, vol. 122, Feb. 2022, doi: 10.1016/j.patcog.2021.108305.
- [10] T. Li, A. Rezaeipناه, and E. S. M. Tag El Din, "An ensemble agglomerative hierarchical clustering algorithm based on clusters clustering technique and the novel similarity measurement," *Journal of King Saud University - Computer and Information Sciences*, Jun. 2022, doi: 10.1016/j.jksuci.2022.04.010.
- [11] T. Niu *et al.*, "Study of degradation of fuel cell stack based on the collected high-dimensional data and clustering algorithms calculations," *Energy and AI*, vol. 10, p. 100184, Nov. 2022, doi: 10.1016/j.egyai.2022.100184.
- [12] Q. Zhao and C. Li, "A new point-infilling and level-setting algorithm enhanced by heuristic searching and incremental clustering to solve global optimization problems," *Journal of King Saud University - Computer and Information Sciences*, Aug. 2022, doi: 10.1016/j.jksuci.2022.08.008.
- [13] D. Minic, "Russia's Invasion of Ukraine A Political-Strategic Break?," Paris, French, Mar. 2022.
- [14] M. Grawjesky, "Russia's war on Ukraine: Background," Brussels, Strasbourg, Luxembourg, Jul. 2022. [Online]. Available: <http://epthinktank.eu>

- [15] I. Kusa, “Russia-Ukraine War,” *Pluto Journals*, vol. 19, no. 1, pp. 7–12, 2022, doi: 10.2307/48676292.