METODE CLUSTER MENGGUNAKAN KOMBINASI ALGORITMA *CLUSTER K-PROTOTYPE* DAN ALGORITMA GENETIKA UNTUK DATA BERTIPE CAMPURAN

CLUSTER METHOD USING A COMBINATION OF CLUSTER K-PROTOTYPE ALGORITHM AND GENETIC ALGORITHM FOR MIXED DATA

Rani Nooraeni
Sekolah Tinggi Ilmu Statistik

Masuk tanggal: 05-12-2015, revisi tanggal: 15-01-2016, diterima untuk diterbitkan tanggal: 19-01-2016

Abstrak

Clustering adalah salah metode utama pada *data mining* yang berguna untuk mengeksplorasi data. Membagi suatu data set berukuran besar ke dalam *cluster* yang sehomogen mungkin adalah tujuan dalam metode *data mining*. Salah satu metode clustering konvensional yaitu algoritma *K-Means* efisien untuk dataset berukuran besar dan tipe data numerik tapi tidak untuk data kategorikal. Algoritma *K-Prototype* menghilangkan keterbatasan pada data numerik tapi dapat juga digunakan pada data kategorikal. Namun solusi yang dihasilkan oleh kedua algoritma tersebut merupakan solusi lokal optimal dimana salah satu penyebabnya adalah penentuan pusat *cluster* awal. Untuk menghadapi masalah tersebut maka algoritma genetika menjadi salah satu usulan yang dapat digunakan untuk mengoptimalkan hasil pengclusteran dengan *K-Prototype*. Hasil dari penelitian menunjukkan optimasi pusat *cluster* dengan algoritma genetika berhasil meningkatkan akurasi hasil cluster dengan *K-Prototype*.

Kata kunci : *Data Mining*, Analisis Cluster, Data Campuran, Algoritma *K-Prototype*, Algoritma Genetika

Abstract

Clustering is one of the main methods in data mining that useful to explore the data. One conventional clustering methods namely the K -Means algorithm efficient for large dataset and numeric data types but not for categorical data type. K-prototype algorithm eliminates the limitations of the numerical data but can also be used on categorical data. But the solutions generated by the algorithm is a local optimal solution in which one of the causes is the determination of the initial cluster's center. Deal with these problems, the genetic algorithm was proposed for solving this global optimasitation problem. The results of the study indicate that the cluster's center optimization with genetic algorithm success to improve the accuracy of the results of the cluster with K–Prototype algorithm.

Keywords : *Data Mining, Cluster Analysis, Mixed Data, K-Prototype Algorithm, Genetic Algorithm*

PENDAHULUAN

Clustering merupakan salah satu metode dalam *data mining*. Metode cluster dalam *data mining* berbeda dengan metode konvensional yang biasa digunakan untuk pengelompokkan. Perbedaannya adalah *data mining* memiliki dimensi data yang tinggi yaitu bisa terdiri dari puluhan ribu atau jutaan record dengan puluhan ataupun ratusan atribut. Selain itu pada *data mining* data bisa terdiri dari tipe data campuran seperti data numerik dan kategorikal.

Metode cluster standar hierarki dapat menangani data bertipe campuran numerik dan kategorikal tetapi ketika data berukuran besar maka timbul masalah dalam hal efisiensi waktu penghitungan. K-Means dapat diterapkan pada data berukuran besar tetapi efisien untuk data bertipe numerik. Hal ini disebabkan pada K-means optimasi *cost function* menggunakan jarak euclidean yang mengukur jarak antara data poin dengan rata-rata cluster. Meminimalkan fungsi *cos*t dengan menghitung rata-rata cocok digunakan untuk data numerik.

Salah satu dataset yang terdiri dari ratusan atribut adalah data hasil Pendataan Potensi Desa (PODES). Data PODES

menyediakan data potensi/keadaan pembangunan hingga level terendah yaitu desa/kelurahan. Data PODES meliputi informasi mengenai keadaan sosial, ekonomi, sarana dan prasarana, serta potensi yang dimiliki suatu desa/kelurahan. Unit observasi dalam PODES adalah desa yang merupakan level wilayah terendah, sedangkan atribut yang terkandung dalam PODES jika dirinci jumlahnya maka jumlahnya mencapai 593 atribut dengan jumlah record mencapai 77.961 unit desa/kelurahan (Tabel 1 (Lampiran 1)).

Berdasarkan karakteristik tersebut maka struktur data PODES merupakan struktur data kompleks. Untuk mengeksplorasi data kompleks diperlukan suatu metode yang tepat sehingga dapat diperoleh hasil yang lebih akurat dengan informasi yang mendalam dan berharga sekaligus dapat menjadi suatu pengetahuan baru dan bermanfaat. Metode analisis yang dapat diterapkan pada kasus tersebut adalah metode analisis *data mining*.

Struktur dataset PODES sejalan dengan realita data yang tersedia dalam kehidupan sehari-hari, dimana data yang tersedia tidak hanya terdiri dari data numerik saja atau data kategorikal saja namun terdapat juga data bertipe campuran. Tabel 1 memperlihatkan struktur dataset PODES secara rinci. Begitu juga dengan data hasil sensus/survey biasanya memiliki tipe data campuran. Hal ini dikarenakan tidak semua persoalan atau pertanyaan bisa dijawab dengan suatu nilai berskala ukur. Oleh karena itu, diperlukan suatu metode analisis yang dapat digunakan untuk menganalisis data bertipe campuran.

Teknik analisis yang dapat menggambarkan karakteristik sekelompok wilayah berdasarkan satu atau lebih variabel salah satunya adalah teknik *clustering*. Dengan teknik clustering akan diperoleh kelompok-kelompok desa, dimana setiap desa yang berada dalam satu kelompok memiliki karakteristik yang mirip dan dengan desa pada kelompok lain sangat berbeda karakteristiknya. Dengan teknik tersebut dapat mempermudah dalam melihat profil suatu desa berdasarkan variabel ciri yang mendominasinya. Dan dengan teknik ini juga mempermudah pengguna data untuk melihat perbandingan karakteristik suatu desa terhadap desa lainnya.

Salah satu metode *clustering konvensional* yang biasa digunakan dalam teknik pengclusteran dan efisien digunakan pada data berukuran besar adalah algoritma *K-Means.* Akan tetapi metode ini cocok untuk data yang bertipe numerik dan tidak efektif jika digunakan untuk tipe data kategorikal. Hal ini dikarenakan cost function yang dihitung menggunakan jarak euclidean hanya cocok untuk data bertipe numerik (Jayaraj, 2014).

Menghadapi kendala tersebut Huang mengusulkan sebuah algoritma yang disebut dengan algoritma *K-Prototype*, untuk menangani masalah *clustering* dengan data bertipe campuran numerik dan kategorikal. *K-Prototype* adalah salah satu metode *clustering* yang berbasis *partitioning*. Algoritma ini merupakan hasil pengembangan dari algoritma cluster *K-Means* untuk menangani *clustering dengan* atribut data bertipe campuran numerik dan kategorikal. *K-Protoype* memiliki keunggulan karena algoritmanya yang tidak terlalu kompleks dan mampu menangani data yang besar serta lebih baik dibandingkan dengan algoritma yang berbasis hierarki (Huang, 1997). Algoritma K-prototype ini telah mendasari banyak penelitian yang menghadapi data besar bertipe campuran seperti penelitian yang dilakukan oleh D.T. Pham (2011), Jengyou He (2011).

Namun demikian baik metode K-Means maupun K-Prototype menghasilkan solusi yang lokal optimum. Kedua metode tersebut sensitif terhadap penentuan inisialisasi posisi pusat cluster dan cenderung mengalami *konvergensi prematur* sehingga menghasilkan solusi optimum lokal akibatnya hasil peng*cluster*an bisa berbeda jika menggunakan inisial pusat cluster random yang berbeda (Dash & Dash, 2012) dan (Feng & Wang, 2011). Begitu juga Duc truong Pham, 2011 dalam penelitiannya mengatakan bahwa proses algoritma K-Means dan K-Prototype seringkali

konvergen pada lokal minimum dan bukan pada global minimum.

Dalam penelitiannya, Huang mengusulkan untuk menerapkan teknik pengoptimasi yang dapat mengatasi masalah optimum lokal, salah satunya dengan menerapkan algoritma genetika (Huang, 1997). Algoritma genetika (GA) merupakan suatu alat optimasi yang dapat digunakan untuk mengotimalkan hasil dari suatu metode. GA dapat digunakan untuk mengoptimalkan inisial center cluster sehingga dapat diperoleh hasil pengclusteran yang global optimum.

Penelitian lainnya yang menunjukkan efektifitas GA dalam pengclusteran misalnya penelitian yang dilakukan Li Jie (2003) menerapkan algoritma genetika dalam pengclusterannya, dan menghasilkan kesimpulan bahwa algoritma genetika efektif dalam menangani data yang kompleks baik dari sisi jumlah record maupun dari jumlah cluster. Begitupula dalam penelitian Rajashree Dash (2012), pengclusteran dengan algoritma genetika menghasilkan cluster yang lebih optimal dibandingkan algoritma K-Means. Sedangkan Dianhu Cheng (2014) mengkombinasikan antara algoritma K-Means dengan algoritma genetika untuk menggabungkan kelebihan dari kedua metode tersebut untuk memperoleh jumlah cluster yang optimal.

Berdasarkan berbagai macam manfaat tersebut maka selanjutnya algoritma genetika akan digunakan dalam penelitian ini untuk mengoptimasi algoritma *K-prototype* yang dapat melakukan pengclusteran pada data bertipe campuran.

Maksud penelitian ini adalah menerapkan metode cluster algoritma *k-prototype* yang dioptimalkan dengan algoritma genetika pada data beratribut campuran dengan tujuan memperoleh solusi optimum global sehingga hasil peng-*cluster*-an menjadi lebih baik dan akurat.

Manfaat yang diharapkan dari penelitian ini adalah kontribusi dalam bidang keilmuan berkaitan dengan pengelompokan wilayah dengan atribut bertipe campuran sehingga dapat menghasilkan kelompok-kelompok desa yang lebih baik, lebih erat kesamaan karakteristiknya, dan lebih akurat dengan kombinasi algoritma *cluster K-Prototype* dan algoritma genetika. Sehingga akan bermanfaat untuk berbagai kalangan baik pemerintah, akademisi dan masyarakat luas dalam memperoleh gambaran/karakteristik dan kondisi suatu wilayah hingga level desa yang lengkap dan akurat dan mendukung dalam menentukan suatu kebijakan agar lebih tepat sasaran.

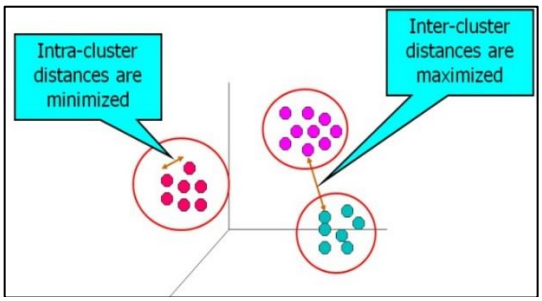
METODOLOGI

Tinjauan Referensi

Clustering merupakan salah satu metode utama pada *data mining*. Tipe data yang dapat dikerjakan dengan metode *data mining* adalah data kompleks, yaitu data yang terdiri dari puluhan ribu atau ratusan ribu *record* dan puluhan atau ratusan atribut. Terdapat berbagai algoritma pengclusteran konvensional yang umum digunakan namun salah satu algoritma yang efisien untuk data berukuan besar adalah algoritma *K-Means*. Keterbatasa *K-means* terhadap data numerik dikembangkan menjadi algoritma *K-prototype*. Masalah dari algoritma *K-Means* dan Algoritma *K-Prototype* adalah hasil ukuran similaritas yang optimum lokal (Huang, 1997). Untuk mengatasi masalah tersebut telah dilakukan pengembangan metode oleh para peneliti. Gambar 1 menunjukkan peta penelitian yang telah dilakukan oleh peneliti lain terkait dengan algoritma *clustering* konvensional, algoritma *clustering* untuk data campuran, algoritma genetika, dan kombinasi antar algoritma yang menjadi rujukan penelitian ini (Gambar 1 (Lampiran 2)).

Clustering adalah pengelompokkan sekumpulan objek yang mirip dengan properti yang sama dalam satu kelompok dan tidak mirip terhadap objek di kelompok lainnya. *Clustering* dan *Classification* adalah dua teknik utama dalam *data mining* yang diikuti kemudian

oleh *association rules*, prediksi, estimasi, dan regresi. (Han & Kamber, 2006), *Clustering* dikenal sebagai *unsupervised learning* karena tidak terdapat informasi label kelas sehingga *clustering* merupakan *learning by obseravtion* daripada *learning by examples.*



Gambar 2. Ilustrasi *Clustering*

Metode analisis cluster membutuhkan suatu ukuran ketakmiripan (jarak) yang didefinisikan untuk setiap pasang objek yang akan dikelompokan. Jarak yang biasa digunakan dalam analisis penggerombolan diantaranya:

1) Ukuran Data numerik, ukuran yang umum digunakan untuk data bertipe numerik adalah ukuran jarak euclidean, sedangkan ukuran lainnya adalah mahalanobis, Manhatan, minkowski, chebyshev dan lain-lain.
2) Ukuran untuk data kategorikal, terdapat ukuran rasio ketidakcocokan, Goodal3 similarity, gambaryan similaruty, dan lain-lain.

Ukuran Kesamaan (*Similarity Measure*)

Bentuk umum ukuran kesamaan dinyatakan sebagai berikut

$$d(X_i, Z_l) = \sum_{j=1}^{m} \delta(x_{ij}, z_{lj}) \quad (1)$$

$z_l = [z_{i1}, z_{i2}, \dots, z_{im}]^T$ adalah prototype untuk cluster l. Ukuran kesamaan untuk atribut numerik dikenal dengan jarak euclidean ditunjukkan dalam persamaan (2) berikut ini

$$d(X_i, Z_l) = \left(\sum_{j=1}^{m_r} \left(x_{ij}^r - z_{lj}^r\right)^2\right)^{1/2} \quad (2)$$

${x_{ij}}^r$ adalah nilai pada atribut numeric j, ${z_{lj}}^r$ adalah rata-rata atau prototype atribut numerik ke j cluster l. m_r adalah jumlah atribut numerik.

Sedangkan ukuran kesamaan untuk data kategorikal adalah

$$d(X_i, Z_l) = \gamma_l \sum_{j=l+1}^{m_c} \delta\left(x_{ij}^c, z_{lj}^c\right) \quad (3)$$

Dimana *simple matching similarity measure* untuk data kategorikal adalah

$$\delta\left(x_{ij}^c, z_{lj}^c\right) = \begin{cases} 0 & \left(x_{ij}^c = z_{lj}^c\right) \\ 1 & \left(x_{ij}^c \neq z_{lj}^c\right) \end{cases} \quad (4)$$

γ_l adalah bobot untuk atribut kategori pada cluster l yang nilainya merupakan nilai standar deviasi untuk atribut numerik pada masing-masing cluster. ketika ${x_{ij}}^c$ adalah nilai atribut kategorikal , z_{lj}^c adalah modus atribut ke j cluster l. m_c adalah jumlah atribut kategorikal.

He, memodifikasi *simple matching similarity measure* menjadi persamaan (5) untuk meningkatkan kemiripan objek dalam cluster dengan atribut kategorikal sehingga hasil pengclusteran menjadi lebih baik. Jika

$$\delta\left(x_{ij}^c, z_{lj}^c\right) = \begin{cases} 1 - \omega(x_{ij}^c, l) & \left(x_{ij}^c = z_{lj}^c\right) \\ 1 & \left(x_{ij}^c \neq z_{lj}^c\right) \end{cases} \quad (5)$$

$\omega(x_{ij}^c, l)$ adalah nilai penimbang untuk x_{ij}^c dimana nilai $\omega(x_{ij}^c, l)$ adalah

$$\omega\left(x_{ij}^c, l\right) = \frac{f\left(x_{ij}^c | C_l\right)}{|C_l| f\left(x_{ij}^c | D\right)} \quad (6)$$

$f\left(x_{ij}^c | C_l\right)$ adalah frekuensi nilai x_{ij}^c dalam kluster l, dan $|C_l|$ adalah jumlah objek dalam kluster l, dan $f\left(x_{ij}^c | D\right)$ adalah frekuensi nilai x_{ij}^c pada keseluruhan dataset. Pada paper ini *matching similarity measure* yang digunakan untuk data kategorikal menggunakan formula He.

Berdasarkan persamaan (1)-(5), maka ukuran kesamaan untuk data yang memiliki atribut numerik dan atribut kategorikal adalah [*2*]

$$d(X_i, Z_l) = \left(\sum_{l=1}^{m_r} \left(x_{ij}^r - z_{lj}^r\right)^2 + \gamma_l \sum_{j=l+1}^{m_c} \delta\left(x_{ij}^c, z_{lj}^c\right)\right)^{1/2} \quad (7)$$

Huang Cost Function

Huang menyatakan persamaan *cost function* untuk data campuran numerik dan kategorikal adalah

$$Cost_l = \sum_{l=1}^{k} u_{il} \sum_{j=1}^{m_r} \left(x_{ij}^r - z_{lj}^r\right)^2 + \gamma_l \sum_{j=1}^{m_c} u_{il} \sum_{j=1}^{m_c} \delta\left(x_{ij}^c, z_{lj}^c\right) \quad (8)$$
$$Cost_l = Cost_l^r + Cost_l^c$$

dimana $Cost_l^r$ adalah biaya total untuk semua atribut numerik dari *object* dalam *cluster l*. $Cost_l^r$ diminimalkan jika z_{jl} dihitung dengan persamaan (9) berikut ini.

$$z_{lj} = \frac{1}{n_l} \sum_{i=1}^{n} u_{il} x_{ij} \quad (9)$$

untuk $j = 1, \ldots, m$

dimana $n_l = \sum_{i=1}^{n} u_{il} x_{ij}$ adalah jumlah *object* di dalam *cluster l.*

Pada atribut kategorikal misalkan C_j adalah sekumpulan nilai unik yang terdapat dalam atribut kategorikal *j* dan $p(c_j \in C_j|l)$ adalah probabilitas dari kemunculan nilai c_j di dalam *cluster l.* maka $Cost_l^c$ dalam persamaan (5) bisa ditulis ulang menjadi

$$Cost_l^c = \gamma_l \sum_{j=1}^{m_c} n_l \left(1 - p\left(q_{jl}^c \in C_j|l\right)\right) \quad (10)$$

dimana n_l adalah jumlah *object* di dalam *cluster l.* Solusi untuk meminimalisasi $Cost_l^c$ dijelaskan dengan Lemma 1 berikut.

Lemma 1: untuk sebuah *cluster* khusus *l*, $Cost_l^c$ diminimalisasi jika dan hanya jika $p\left(z_{lj}^c \in C_j|l\right) \geq p\left(c_j \in C_j|l\right)$ untuk $z_{lj}^c \neq c_j$ untuk semua atribut kategorikal. Akhirnya *Cost* bisa dituliskan ulang dengan

$$Cost = \sum_{l=1}^{k} (Cost_l^r + Cost_l^c) = \sum_{l=1}^{k} Cost_l^r + \sum_{l}^{k} Cost_l^c = Cost^r + Cost^c \quad (11)$$

Persamaan (10) adalah *cost function* untuk *Clustering dataset* dengan atribut bernilai numerik dan kategorikal. Karena $Cost^r$ dan $Cost^c$ adalah non-negatif, minimalisasi *Cost* bisa dilakukan dengan meminimalkan $Cost^r$ dan $Cost^c$, *total cost* dari atribut numerik dan kategorikal untuk semua *cluster*.

Algoritma K-Prototype

Algoritma *K-Prototype* adalah salah satu metode *Clustering* yang berbasis *partitioning*. Algoritma ini adalah hasil pengembangan dari algoritma *K-Means* (Huang,1998) untuk menangani c*lustering* pada data dengan atribut bertipe campuran numerik dan kategorikal. Pengembangan yang dilakukan oleh Huang mempertahankan efisiensi algoritma K-Means dalam menghadapi data berukuran besar dan dapat diterapkan pada data numerik dan kategorikal. Pengembangan yang mendasar pada algoritma K-Prototype terdapat pada pengukuran kesamaan (*similarity measure*) antara *object* dengan *centroid (prototype)*-nya.

Secara umum algoritma K-Prototype terbagi kedalam tiga tahapan utama (Huang 1997), yaitu:

1. Inisialiasi awal *prototype*. Pada proses ini akan dilakukan pemilihan sejumlah *k prototype* secara acak dari *dataset X* sesuai dengan jumlah *cluster* yang ditentukan.
2. Alokasi objek di dalam *X* ke *Cluster* dengan *prototype* terdekat. Ukur Jarak Objek ke semua *prototype* dan tempatkan objek pada *cluster* terdekat. Tahap ini algoritma *K-Prototype* mengalokasikan semua *object* didalam *dataset* ke *cluster* dimana *prototype* dari *cluster* tersebut memiliki jarak yang paling dekat ke *object* data. Pengalokasian semua *object* di dalam daset *X* ke *cluster* yang memiliki jarak *prototype* terdekat dengan *object* yang diukur. Untuk setiap kali *object X* selesai dialokasikan, maka selanjutnya akan dilakukan penghitungan *(update)* terhadap *prototype cluster* yang berkaitan.
3) Realokasi *object* Jika terjadi perubahan *prototype*. Setelah semua *object* dalam X selesai dialokasikan, selanjutnya akan dilakukan pengukuran ulang jarak antara semua *object* di dalam X terhadap semua *prototype* yang ada. Jika ditemukan adanya *object* yang ternyata lebih dekat ke *prototype* yang lain, maka akan dilakukan pemindahan

keanggotaan dan kemudian akan dilakukan update terhadap *prototype cluster* lama dan *prototype cluster* baru. Proses ini akan terus dilakukan sampai tidak ada lagi perubahan *prototype* atau sampai kriteria *stopping* terpenuhi.

Evaluasi Hasil Cluster

Metode yang umum digunakan untuk mengukur hasil pengclusteran dengan data campuran adalah *Total Cost* dan *Categorical Variance Criterion* (CVC) (Hsu & Huang, 2008). CVC menggabungkan antara metode *Category Utility* dan pengukuran variance untuk data numerik. Semakin besar nilai CVC maka semakin bagus juga hasil *clustering.* Persamaan CVC sebagai berikut:

$$CVC = CU/(1+\sigma^2) \qquad (12)$$

Fungsi *Categorical Utility* (CU) bertujuan untuk memaksimalkan kemungkinan atau probabilitas bahwa dua buah object di dalam cluster yang sama memiliki nilai atribut yang sama dan probabilitas bahwa dua object pada cluster yang berbeda memiliki atribut yang berbeda. *Categorical utility* untuk sebuah dataset dapat dihitung sebagai berikut:

$$CU = \sum_l \left(\frac{|C_l|}{|D|} \sum_j \sum_i \left[P(A_j = V_{ij} \mid C_l)^2 - P(A_j = V_{ij})^2 \right] \right) \qquad (13)$$

$P(A_j = Vij \mid C_l)$ adalah probabilitas kondisional dimana atribut *j* memiliki nilai V_{ij} di dalam cluster C_l, dan $P(A_j = V_{ij})$ probabilitas keseluruhan bahwa atribut *j* memiliki nilai V_{ij} di seluruh dataset.

Variance (σ^2), bisa digunakan untuk mengevaluasi kualitas *clustering* untuk data numerik. Total *variance* dapat diperoleh dengan melakukan penambahan semua *variance* di setiap cluster, dimana pada setiap *cluster* akan dilakukan penambahan *variance* dari setiap data numerik. Persamaannya sebagai berikut:

$$\sigma^2 = \sum_l \frac{1}{|C_l|} \sum_j \sum_i (V_{ij}^l - V_{j,avg}^l)^2 \qquad (14)$$

Dalam hal ini, V_{ij}^l dan $V_{j,avg}^l$ adalah record ke-*i* dan nilai rata-rata atribut numerik ke-*j* pada cluster ke C_l.

Algoritma Hibrida K-Prototype-GA

Kim dkk (2008), menggunakan kombinasi algoritma *K-Means* dengan algoritma Genetika pengelompokan pelanggan dalam membuat *recomender system* pada *online shoping market.* Dari hasil penelitian tersebut, bisa disimpulkan bahwa *GA-K-Means* mampu menghasilkan pengelompokan *(clustering)* yang lebih baik dibandingkan dengan *Self Organising Map* (SOM) yang berbasis *Neural Network.*

Min Feng melakukan penelitian untuk mengoptimalkan Algoritma *K-Means* dalam menentukan pusat awal *cluster,* dimana hasil penelitian menunjukkan bahwa algoritma *K-Means* memiliki kelemahan tidak hanya memiliki ketergantungan pada data awal (*inisial center cluster*), tetapi juga konvergensi yang cepat (konvergensi prematur) dan hasil *clustering* yang kurang akurat (Feng & Wang, 2011). Untuk memperoleh *cluster* yang efektif dan akurat maka Min Feng dan Zhenyan-wang mengoptimalkan Algoritma *K-Means* (PKM) dengan Algoritma Genetika menjadi sebuah Algoritma Hibrid (PGKM). Percobaan menunjukkan bahwa algoritma ini dapat mengatasi masalah pada penentuan *inisial center cluster,* konvergensi premature dan waktu pengolahan yang lebih efisien.

Hasil penelitian Liu dkk, tahun 2008 menggunakan algoritma genetik yang dikombinasikan dengan algoritma *K-Means* untuk menemukan variabel yang valid dan jumlah *cluster* optimal secara simultan. Hasil penelitian menunjukkan, metode hibrid tersebut behasil menghilangkan variabel yang tidak relevan dan menghasilkan jumlah *cluster* secara otomatis, dan berhasil meningkatkan hasil pengelompokan pelanggan secara signifikan (Liu & Ong, 2008).

Berdasarkan pada fakta-fakta yang didapatkan dari beberapa penelitian sebelumnya maka penulis melalui penelitian ini mengusulkan untuk mengkombinasikan algoritma genetika dengan metode clustering yang diusulkan dalam penelitian ini yaitu algoritma *K-*

Prototype untuk memperoleh tingkat akurasi yang lebih baik.

Dalam penelitian ini metode algoritma genetika digunakan untuk memperoleh inisial *center cluster* yang optimal. Tahapan algoritma metode hibrida *K-prototype*-GA dalam penelitian ini adalah sebagai berikut:

1) Menentukan inisial populasi, meliputi jumlah gen dalam kromosom dan jumlah kromosom dalam individu
2) Melakukan proses algoritma K-Prototype untuk setiap kromosom dalam populasi.
3) Menghitung nilai *fitness* dari tiap kromosom dalam populasi berdasarkan nilai *cost function*.
4) Memilih kromosom berdasarkan nilai *fitness*.
5) Melakukan perkawinan silang (*crossover*) dan mutasi untuk mendapatkan keturunan (*offspring*).
6) Melakukan *elitism* dan *replacement* sehingga diperoleh populasi baru.
7) Kembali ke langkah 2 hingga kriteria yang ditentukan terpenuhi
8) Setelah diperoleh hasil dari proses algoritma genetika kemudian digunakan untuk proses *clustering* dengan algoritma *K-Prototype*.

Metode Analisis

Algoritma Genetika (GA) banyak digunakan dalam masalah pencarian parameter optimal, dengan demikian algoritma genetika akan digunakan untuk mengoptimalkan hasil pengclusteran dengan *K-prototype*. Penerapan GA dalam penelitian ini memiliki fungsi untuk menghasilkan inisial *center cluster* yang optimal sehingga pada tahap awal inisialisasi populasi dengan *K-Prototype* menggunakan *center cluster* dari hasil pencarian Algoritma Genetika (Feng & Wang, 2011).

Preprocessing

Sebelum melakukan pemodelan akan dilakukan preprocessing, yang pertama pemeriksaan data missing, yang kedua tranformasi data numerik. Pada variabel penelitian ini terdapat dua variabel kategorikal yang memiliki missing value yaitu variabel permukaan jalan dan variabel kondisi jalan yang dapat dilalui kendaraan beroda empat atau lebih. Missing terjadi karena terdapat desa yang tidak memiliki sarana transportasi darat oleh sebab itu penulis mengganti nilai missing kategori tersebut dengan kategori tidak memiliki transportasi darat.

Transformasi yang dilakukan adalah melakukan standardisasi data numerik menjadi Z-Score. Hal ini dilakukan karena data memiliki satuan yang berbeda. Proses standarisasi menjadikan dua data dengan perbedaan satuan yang lebar akan otomatis menjadi menyempit (Santoso,2010). Dengan demikian analisis perbandingan antar variabel pun dapat dibandingkan.

Selain standarisasi dan transformasi, akan dilakukan juga pembuatan *look up table* untuk mengefisienkan waktu penghitungan jarak pada saat menjalankan algoritma genetika. *Look up table* merupakan tempat yang menyimpan semua jarak antar objek, strategi ini digunakan untuk mengefisienkan waktu computing seperti yang dilakukan oleh Lin dan Yang (2005).

Kemudian hal lain yang perlu dipersiapkan pada saat akan melakukan proses clustering adalah menyiapkan beberapa inputan kategori sebagai berikut:

1) Ukuran Populasi. Ukuran populasi diperlukan pada saat akan mengeksekusi algoritma genetika. Tidak ada ketentuan dalam menentukan ukuran populasi. Jika jumlah kromosom yang digunakan terlalu sedikit, maka individu yang dapat digunakan untuk proses *crossover* dan mutasi akan sangat terbatas, sehingga menyia-nyiakan proses yang ada. Sedangkan jika jumlah kromosom yang digunakan terlalu banyak, akan memperlambat proses algoritma genetika yang dilakukan. Semakin besar ukuran populasi dalam satu generasi, maka akan menghasilkan solusi yang lebih baik. Dalam penelitian ini ukuran populasi adalah 1000 kromosom untuk

fungsi inisial center dan 50 untuk fungsi penentuan variabel relevan.

2) Generasi Maksimal. Generasi maksimal atau iterasi maksimal mempengaruhi jumlah komputasi yang dilakukan pada saat pengolahan data, dimana 1 generasi dapat mewakili sebesar populasi yang telah ditentukan. jika terdapat 10 generasi dan ukuran populasi 1000 maka komputasi yang dilakukan akan sebanyak 10000 kali sehingga menghasilkan kromosom yang lebih variatif dalam proses *fitness*. Kromosom menjadi variatif dikarenakan pada saat pencarian nilai *fitness* terbaik pada generasi pertama telah selesai, maka selanjutnya dilakukan proses pindah silang dari populasi yang ada, sehingga pada generasi selanjutnya akan menghasilkan populasi yang baru. Pada penelitian ini generasi maksimal yang ditetapkan adalah 100.
3) Jumlah *Cluster*. Jumlah *Cluster* (*K*) ditentukan di awal untuk menggelompokkan data yang diolah sesuai dengan jumlah *cluster* yang diinginkan. Jumlah cluster minimal (*Kmin*) adalah 2 dan maksimal (*Kmaks*) adalah *n/2* atau $\sqrt{n}$ (Lin, Yang, & Kao, 2005) dimana *n* adalah jumlah data. Kemudian jumlah cluster bisa ditentukan secara random dengan rentang [*Kmin, Kmaks*]. Dalam penelitian ini jumlah *cluster* awal ditetapkan berdasarkan hasil pengamatan perubahan nilai cost function pada saat nilai *K=2* sampai dengan *K=20*.
4) Parameter algoritma genetika yang disarankan De Jong (A.A., 2001) dalam (Zukhri, 2013) adalah (1) Probabilitas penyilangan cukup besar lebih dari 50%, (2) Probabilitas mutasi cukup kecil (sebuah gen untuk sebuah kromosom), (3) Ukuran populasi berkisar antara 50 sampai 500 kromosom. Walaupun demikian tidak ada batasan yang pasti mengenai besaran nilai probabilitas crossover, mutasi dan ukuran kromosom tergantung dari tujuan penelitian. Dengan mengacu ketetapan De Jong maka batas probabilitas penyilangan yang digunakan pada penelitian ini adalah 0,5, probabilitas mutasi adalah 0,1 dan ukuran populasi untuk proses penentuan center cluster optimal dan pemilihan variabel relevan berturut-turut adalah 1000 dan 50.

Pemodelan

K-Prototype

Pada proses *Clustering* dengan *K-Prototype* dilakukan beberapa proses utama yang terbagi kedalam 3 tahapan utama sebagai berikut (Huang, 1997):

1) Inisialiasi awal *prototype*. Pada proses ini akan dilakukan pemilihan sejumlah k *prototype* secara acak dari *dataset* X sesuai dengan jumlah *cluster* yang ditentukan.
2) Alokasi objek di dalam X ke *Cluster* dengan *prototype* terdekat. Berikut *pseudocode* dari algoritma alokasi objek kedalam *cluster* pada *K-Prototype* (Gambar 3 (Lampiran 3)):
3) Realokasi *object* Jika terjadi perubahan *prototype*. Proses ini akan terus dilakukan sampai tidak ada lagi perubahan *prototype* atau sampai kriteria *stopping* terpenuhi. Berikut *pseudocode* algoritmanya (Gambar 4 (Lampiran 4)):

K-Prototype-GA* untuk Optimasi Inisial *Center Cluster

Desain dari optimasi yang dilakukan pada *K-Prototype* terletak pada inisial pusat *cluster*. Jadi ketika pusat *cluster* di optimasi dengan Algoritma Genetik diharapkan *K-Prototype* mempunyai awalan *prototype* yang bagus, sehingga untuk proses selanjutnya dapat memperoleh *cluster* yang lebih akurat. Alur proses penggabungan metode K-Prototype dan Genetika untuk memperoleh center cluster optimal dalam penelitian ini dapat dilihat pada Gambar 5 (Lampiran 5).

1) Inisialisasi Populasi Awal

Penelitian ini melibatkan 37 variabel dan 77.961 record data dalam penelitian ini. Unit record adalah unit desa yang diberikan indeks mulai dari nomor 1 sd 77.961. Fase inisialisasi populasi ini digunakan untuk menentukan sejumlah kromosom awal yang akan digunakan untuk komputasi selanjutnya. Berdasarkan pada penelitian Jie dan Li-Cheng (2003) maka dalam membentuk kromosom individu, panjang gen ini adalah sama dengan jumlah *K* (jumlah *cluster*) dari proses *Clustering*. Dimana masing-masing nilai yang ada pada gen akan mewakili no record data pada proses *Clustering* (Jie, Xinbo, & Li-Cheng, 2003). Jadi nilai yang ada pada gen adalah no ID desa yang terpilih secara acak dari nomor 1-77.961 sebanyak *k* cluster. Jika jumlah kromosom yang akan dibentuk adalah 1000 maka akan ada 1000 x *k* yaitu 1000*k* desa yang akan terpilih pada populasi awal ini.

Dalam penelitian ini jumlah cluster ditetapkan setelah menganalisa grafik perubahan nilai cost function dari $k \in [2,20]$. Jumlah cluster yang paling signifikan penurunan nilai cost functionnya akan dijadikan nilai *k* pada metode pengclusteran dengan *k-prototype*.

2) Evaluasi Fitness

Proses evaluasi dengan alat ukurnya adalah fungsi *fitness* merupakan proses untuk mengevaluasi setiap populasi dengan menghitung nilai *fitness* setiap kromosom dan mengevaluasinya sampai terpenuhi kriteria berhenti. Nilai *fitness* menyatakan nilai dari fungsi tujuan. Tujuan dari algoritma genetika adalah memaksimalkan nilai *fitness*. Berikut formulanya:

$$f = \frac{1}{(h+a)} \quad (15)$$

h adalah suatu nilai yang sangat kecil untuk menghindari pembagian dengan nilai 0. Sedangkan *a* adalah fungsi cost pada *K-Prototype*.

Kemudian untuk menghindari optimum lokal, dibuatlah suatu mekanisme yang disebut dengan *Linier Fitness Rangking* (LFR). Tujuan dari mekanisme ini sebenarnya adalah untuk melakukan penskalaan nilai-nilai *fitness* dengan menggunakan persamaan berikut:

$$LFR(i) = f_{max} - (f_{max} - f_{min}) \left(\frac{R(i)-1}{N-1}\right) \quad (16)$$

Keterangan:

- LFR(i) = nilai LFR individu ke-i
- N = jumlah individu dalam populasi
- R(i) = rangking individu ke-i setelah diurutkan dari nilai fitness terbesar hingga terkecil
- fmax = nilai *fitness* tertinggi
- fmin = nilai *fitness* terendah

3) *Elitisme* dan *Replacement*

Proses elitisme diperlukan untuk mencegah kehilangan solusi terbaik, tahap ini dapat meningkatkan performansi algoritma genetika secara cepat. Pada saat membuat populasi baru dengan kawin silang dan mutasi, kromosom terbaik dapat hilang. Elitism adalah metode untuk mengganti kromosom terjelek dengan kromosom terbaik.

Individu terbaik ini ditentukan berdasarkan nilai fitnessnya, individu/kromosom dirangking berdasarkan besaran nilai fitnessnya. Semakin besar nilai fitnessnya senakin baik kromosom/individu tersebut. penentuan individu terbaik dilakukan untuk kebutuhan proses crossover dan mutasi. Jika ukuran populasi adalah ganjil maka kromosom terbaik di copy sebanyak satu kromosom, sedangkan jika ukuran populasi genap maka kromosom yang dicopy sebanyak dua kromosom. Kromosom terbaik ini akan dibandingkan dengan kromosom hasil penyilangan jika nilainya lebih besar dari hasil penyilangan maka copy kromosom elit kedalam iterasi berikutnya. Jika lebih kecil maka replace kromosom elit dengan kromosom terbaik hasil penyilangan.

4) Seleksi

Seleksi dilakukan dalam rangka untuk mendapatkan calon induk yang baik

yang akan menjalani proses *crossover* dan *mutasi*. Metode yang banyak digunakan dalam proses seleksi adalah teknik *roullete wheel*. Pendekatan ini dilakukan dengan menghitung nilai probabilitas seleksi (p) tiap individu/kromosom berdasarkan nilai fitnessnya dengan persamaan sebagai berikut:

$P_i = \frac{f_i}{f_{total}}$ i=1,2, ..., *pop size*

f_i menyatakan nilai *fitness* dari individu ke-i dan f_{total} adalah total nilai *fitness* dari semua individu.

Setelah diperoleh nilai *p* kemudian dihitung *probabilitas kumulatif* yang akan digunakan pada proses seleksi tiap individu. Kemudian untuk memilih tiap individu bangkitkan nilai peluang *r* secara random. Pilih individu yang nilai probabilitas kumulatifnya $p_{kum} \geq r$.

5) Proses Penyilangan (*Crossover*)

Crossover adalah operator dalam algoritma genetika untuk melakukan operasi pertukaran gen-gen yang bersesuaian dari dua induk untuk membentuk individu baru. Proses perkawinan silang dilakukan berdasarkan probabilitas kawin silang yaitu $P_c \in [0,1]$. Dibangkitkan suatu bilangan random p untuk menentukan terjadi kawin silang atau tidak. Apabila $p \geq P_c$ maka tidak terjadi kawin silang. Menurut De Jong nilai Pc disarankan untuk ditetapkan cukup besar berkisar 50% sampai 70% (A.A., 2001) dalam (Zukhri, 2013). Kemudian untuk menentukan titik potong maka dilakukan juga dengan membangkitkan suatu bilangan acak [1,panjang gen-1].

6) Mutasi

Mutasi dilakukan untuk mencegah algoritma berada pada optimum lokal. Mutasi merupakan proses menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inisialisasi populasi. Mutasi juga terjadi pada probabilitas tertentu yaitu $P_m \in [0,1]$. Pada tahap ini pada setiap gen dibangkitkan suatu bilangan *p*, jika *p* lebih kecil dari p_m yang ditetapkan maka gen tersebut akan dikenai proses mutasi. Proses menggantikan nilai dalam gen yang terkena mutasi terdapat beberapa cara. Pertama dengan membangkitkan bilangan acak dari separuh jumlah record. Kedua menukar nilai pada gen tersebut dengan gen lain yang juga terkena mutasi.

Setelah proses crossover dan mutasi selesai maka akan dilakukan kembali proses evaluasi dengan menghitung fitness dan membandingkan dengan kromosom elite. Kemudian dilakukan replacement jika kromosom baru lebih baik dibandingkan kromosom elite. Begitu seterusnya hingga kriteria berhenti terpenuhi.

Data Penelitian

Dalam penelitian ini, penulis akan melakukan studi kasus menggunakan dataset PODES 2011 se-Indonesia. Dataset yang digunakan dalam penelitian ini terdiri dari 77.961 *record* yang menunjukkan 77.961 desa dan 71 atribut yang dikelompokkan menjadi 37 variabel.

Variabel yang digunakan dalam penelitian ini berdasarkan pada kajian Identifikasi Desa Tertinggal tahun 2002 yang diselenggarakan oleh BPS tahun 2003 menggunakan PODES 2002. Variabel penelitian yang digunakan akan disesuaikan dengan kondisi kuesioner PODES 2011.

HASIL DAN PEMBAHASAN

Hasil *K-Prototype* Tanpa GA

Eksekusi program utama K-Prototype bertujuan untuk mendapatkan pengelompokan dengan nilai total *cost function* terkecil. Total cost menunjukkan total jarak setiap objek terhadap prototype cluster. Semakin kecil nilai total cost maka semakin dekat jarak antara objek dengan prototype clusternya.

Pada Gambar 6 (Lampiran 6) terlihat perubahan nilai total cost dengan beberapa kali percobaan, mulai dari K = 2 dan sampai dengan K = 20. Pada Gambar 6 jumlah cluster dimulai dari dua dengan total nilai cost adalah 1,490438 x 10^6. Semakin besar jumlah cluster yang ditentukan maka nilai total cost semakin mengecil. Penurunan yang paling signifikan adalah pada saat *k* bernilai 4, 6, dan 13, yang mengalami penurunan sebesar 7,44657 x 10^5 , 5,10553 x 10^5 dan 6,63805 x 10^5. Berdasarkan efisiensi waktu pengolahan dan pertimbangan kecukupan jumlah cluster maka nilai *k* yang ditetapkan adalah *k* = 6.

Hasil dari proses clustering dengan K-Prototype, dimana K = 6, maka diperoleh jumlah anggota tiap cluster seperti yang tertera pada Tabel 2.

Pada Tabel 2 diperoleh informasi bahwa anggota *cluster* terbanyak terdapat pada cluster 2 dengan persentase sebesar 41,44 persen. Sedangkan *cluster* 6 memiliki persentase terkecil yaitu 0,08 persen. Jika dilihat dari aspek pemerataan kondisi sosial ekonomi dan prasarana berdasarkan indikator ketertinggalan desa maka desa-desa pada *cluster* 6 merupakan kelompok desa yang sangat berbeda karakteristik sosial ekonominya dibandingkan dengan kelompok besar desa lainnya.

Tabel 2. Jumlah anggota Per Cluster

Cluster i	Jumlah Anggota (Desa)	%
Cluster 1	12929	16,58
Cluster 2	32308	41,44
Cluster 3	28502	36,56
Cluster 4	2364	3,03
Cluster 5	1797	2,30
Cluster 6	61	0,08

Hasil Hibrid *K-Prototype* dengan Algoritma Genetika

Metode hibrid yang berbasis *K-Prototype* merupakan metode untuk membangkitkan inisial pusat *cluster* yang sudah dioptimasi dengan metode algoritma Genetika, kemudian inisial pusat cluster tersebut digunakan dalam melakukan pengclusteran dengan algoritma *K-Prototype*. Sehingga algoritma genetika dalam metode ini hanya digunakan untuk memperoleh calon inisial pusat *cluster* yang baik.

Tahapan algoritma genetika dalam memperoleh inisal pusat *cluster* terbaik adalah sebagai berikut:

1) Menentukan populasi kromosom awal secara acak. Tahapan menentukan kromosom awal dilakukan pada saat input kategori yang dipilih secara acak sebanyak 1000 kromosom dengan jumlah cluster 6.
2) Evaluasi nilai *fitness*. Evaluasi nilai fitness seluruh kromosom untuk mencari nilai terbaik dari clustering yang dilakukan. Kemudian evaluasi nilai *fitness* terbaik per iterasi dalam setiap jumlah cluster. Hasilnya dapat dilihat pada Gambar 7 (Lampiran 7). Gambar 7 memperlihatkan pergerakan perubahan nilai fitness dari best kromosom per iterasi. Perubahan nilai fitness masih bergerak hingga iterasi ke 14 setelah itu mencapai nilai konvergen pada iterasi ke 15 dan seterusnya dengan nilai fitness adalah 9,4423 x 10^{-8}. Dengan demikian nilai fitness terbaik pada saat jumlah cluster 6 adalah 9,4423 x 10^{-8}. Maka *the best chromosom* adalah sebagai berikut:

180	13158	26151	39048	52138	65201

Hasil dari *the best chromosom* merupakan inisial center cluster pada proses clustering dengan algoritma *k-prototype.* Nilai pada setiap gen adalah no id desa. Panjang gen pada Gambar 8 (Lampiran 8) menunjukkan jumlah cluster.

Nomor objek pada Gambar 7 dan 8 akan menjadi inisial center pada proses pengclusteran dengan K-Prototype. Persentase jumlah anggota per cluster yang dihasilkan dari algoritma K-Prototype setelah mengoptimasi inisal centernya dapat dilihat pada Tabel 3.

Tabel 3. Jumlah Anggota per Cluster

Cluster i	Jumlah Anggota (Desa)	%
Cluster 1	2179	2,79
Cluster 2	16341	20,96
Cluster 3	24630	31,59
Cluster 4	61	0,078
Cluster 5	1789	2,29
Cluster 6	32961	42,28

Evaluasi Perbandingan Hasil Clustering

Baik atau tidaknya *cluster* yang dihasilkan dari kedua model tersebut akan dilihat dari beberapa alat ukur yang dapat digunakan untuk data campuran yaitu *Total Cost* dan *Categorical Variance Criterion*. *Total cost* adalah total jarak dari setiap objek ke cluster tempat dia berada. Semakin kecil nilai *total cost* maka cluster yang terbentuk semakin *compact.*. Model yang akan dibandingkan tersebut dapat dilihat pada Tabel 4.

Tabel 4. Perbandingan hasil *Clustering*

No	Model	jumlah cluster	Total Cost	CVC
1	Model K Prototype tanpa Genetika	6	$1204{,}9 \times 10^3$	0,0031
2	Model K Prototype –Genetika untuk center cluster optimal	6	$1202{,}9 \times 10^3$	0,0054

Berdasarkan Tabel 4 maka model K Prototype-Genetika lebih baik dibandingkan dengan model K-Prototype saja tanpa dioptimasi dengan genetika. Ditunjukkan oleh nilai total cost pada K-Prototype yaitu $1204{,}9 \times 10^3$ lebih besar dibandingkan dengan model K-Prototype-Genetika yang menggunakan optimasi dengan genetika yaitu $1202{,}9 \times 10^3$. Nilai CVC pada model K-Prototype-Genetika untuk center cluster optimal merupakan nilai yang terbesar yaitu 0,0054. Semakin besar nilai CVC maka semakin bagus clustering yang dihasilkan. Maka dalam penelitian ini model K-Prototype-Genetika untuk center cluster optimal menghasilkan akurasi cluster yang lebih baik diantara metode lainnya. Artinya dengan kondisi data yang sama model K-Prototype-Genetika untuk center cluster optimal lebih mampu menghasilkan cluster yang lebih homogen dibandingkan model lainnya. Hal ini menunjukan jika model ini lebih baik dibandingkan dengan model lainnya.

Tabel 5. Nilai CU dan Varians pada setiap model penelitian

No	Model	CU	Varians
1	Model K Prototype tanpa Genetika	1,1406	366,7799
2	Model K Prototype –Genetika untuk center cluster optimal	1,1351	209,073

Jika dilihat dari hasil evaluasi dengan melihat nilai *total cost* dan CVC kedua model tersebut, maka perbedaan kedua model terlihat tidak terlalu signifikan. Maka penulis menyarankan pada penelitian selanjutnya dilakukan pembuangan outlier dan pemilihan variable yang relevant terlebih dahulu.

KESIMPULAN DAN SARAN

Metode gabungan *K-Prototype* dengan Algortima Genetika yang diusulkan dalam penelitian ini menghasilkan inisial pusat *cluster yang* optimal. Hal ini terlihat dari hasil percobaan yang telah dilakukan, dimana pada saat dilakukan pengujian menggunakan total cost, model K-Prototype- GA menghasilkan nilai total cost sebesar $1202{,}9 \times 10^3$. Model K-Prototype tanpa GA menghasilkan nilai total cost sebesar $1204{,}9 \times 10^3$. Dengan demikian model K-Prototype-GA untuk kasus penelitian pengelompokan desa berdasarkan indikator ketertinggalan desa menggunakan dataset PODES 2011 memiliki tingkat akurasi yang lebih baik dibandingkan model cluster dengan K-Prototype tanpa Genetika.

Berdasarkan nilai index clustering criterion maka pada kasus penelitian ini model K-Prototype-Genetika untuk center cluster optimal merupakan model terbaik

karena nilainya lebih tinggi dari yang lainnya yaitu 0.0054 dibandingkan 0,0031 dimana hal ini menunjukkan bahwa tingkat kesamaan ciri atau karakteristik dari setiap kelompok yang terbentuk pada model K-Prototype-Genetika untuk optimasi inisial center cluster lebih mirip.

Penelitian ini memanfaatkan metode k-prototype dengan GA sebagai metode utama dalam proses *clustering*. Perlu dilakukan penelitian lebih lanjut untuk dapat menghasilkan clustering yang lebih baik mengingat begitu kompleksnya struktur data dalam penelitian ini serta tipe atribut berupa campuran, numerik dan kategorikal, menyebabkan proses pengolahan semakin komplek dan waktu pengolahan yang panjang.

Untuk mengevaluasi hasil pengelompokan, penulis menyarankan untuk mencari dan menggunakan alat ukur lainnya yang cocok digunakan untuk mengevaluasi hasil pengclusteran dengan data yang bertipe campuran.

DAFTAR PUSTAKA

Amir Ahmaddan Lipika Dey, "A k-mean clustering algoritm for mixed numeric and categorical data," *DAta & Knowledge Engineering*, vol. 63, pp. 503-527, 2007.

BPS, *Metodologi dan Profil Kemiskinan Tahun Tahun 2002*., 2003.

Ch. D. V. Subba Rao, C. Kishore and Shreyash Raju Srinivasulu Asadi, "Clustering the Mixed Numerical and Categorical Datasets Using Similarity Weight and Filter Method," *VSRD International Journal of Computer Science & Information Technology*, vol. 2 (5), pp. 373-385, 2012.

Dharmendra K Roy, Lokesh K Sharma, "Genetic k-Means Clustering Algorithm for Mixed Numeric and Categorical Data Sets," *International Journal of Artificial Intelligece & Applications (IJAIA)*, vol. 1, April 2010.

Gil David, Amir Averbuch, "SpectralCAT:Categorical spectral clustering of numerical and nominal data," *Pattern Recognition*, vol. 45, pp. 416-433, 2012.

J.Han Kamber, *Data Mining Concepts and Techniques*, 2nd ed. San Fransisco, United States of America: Dianne Cerra, 2006.

J. Suguna, M.Arul Selvi, "Ensemble Fuzzy Clustering for Mixed Numeric and Categorical Data," *International Journal of Computer Applications (0975-8887)*, vol. 42 - No 43, Maret 2012.

M. Ramakrishnan, D. Tennyson Jayaraj, "Modified K-Means Algorithm for effective Clustering of Categorical Data Sets," *International Journal of Computer Applications (0975-8887)*, vol. 89 - No 7, Maret 2014.

Ramesh Valaboju, N. Raghava Rao V.N. Prasad Pinisetty, "Hybrid Algorithm for Clustering Mixed Data Sets," *IOSR Jornal of Computer Engineeering (IOSRJCE)*, vol. 6, no. 2, pp. 09-13, Sep-Okt 2012.

Zhexue Huang, "Clustering Large Data Sets with Mixed Numeric and Categorical Values".

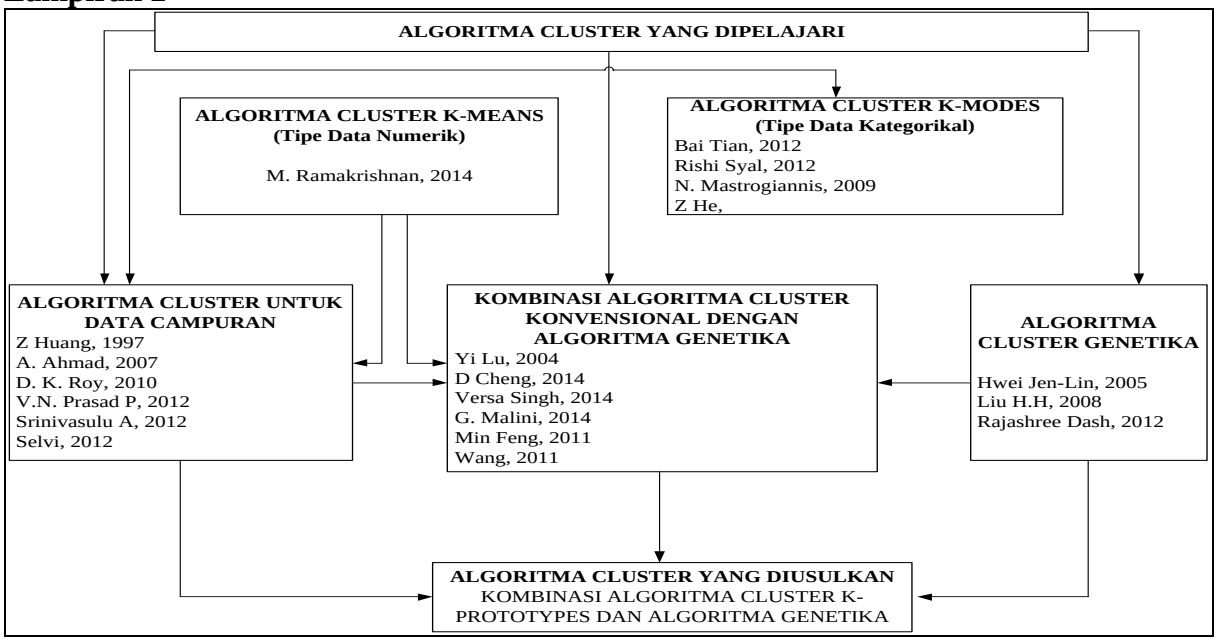
LAMPIRAN

Lampiran 1

Tabel 1. Jumlah atribut kuesioner PODES 2011 menurut bagian-bagian pertanyaan

Bagian kuesioner	Uraian	Jumlah Atribut		
		Numeric	Categorical	Total
BLOK I	Pengenalan tempat	-	15	15
BLOK II	Keterangan petugas	-	4	4
BLOK III	Keterangan umum desa/kelurahan	2	19	21
BLOK IV	Kependudukan dan ketenagakerjaan	7	2	9
BLOK V	Perumahan dan lingkungan hidup	11	38	49
BLOK VI	Bencana alam dan penanganan bencana alam	30	56	86
BLOK VII	Pendidikan dan kesehatan	87	40	127
BLOK VIII	Sosial budaya	24	20	44
BLOK IX	Hiburan dan olahraga	2	19	21
BLOK X	Angkutan, komunikasi dan Informasi	6	31	37
BLOK XI	Penggunaan lahan	6	6	12
BLOK XII	Ekonomi	25	13	38
BLOK XIII	Keamanan	11	56	67
BLOK XIV	Otonomi desa dan program pemberdayaan masyarakat	7	48	55
BLOK XV	Keterangan aparatur desa	2	6	8
Total		**220**	**373**	**593**

Lampiran 2



Gambar 1. Peta Penelitian yang Terkait *Clustering*

Lampiran 3

```
FOR i = 1 TO NumberOfObjects
  Mindistance=       Distance(X[i],O_prototypes[1])+       gamma*
  Sigma(X[i],C_prototypes[1])
   FOR j = 1 TO NumberOfClusters
      distance=     Distance(X[i],O_prototypes[j])+     gamma     *
      Sigma(X[i],C_prototypes[j])
        IF (distance < Mindistance)
          Mindistance=distance
          cluster=j
        ENDIF
   ENDFOR
  Clustership[i]=cluster
  ClusterCount[cluster] + 1
   FOR j=1 TO NumberOfNumericAttributes
      SumInCluster[cluster,j] + X[i,j]
      O_prototypes[cluster,j]=SumInCluster[cluster,j]/ClusterCoun
      t[cluster]
   ENDFOR
   FOR j=1 TO NumberOfCategoricAttributes
      FrequencyInCluster[cluster,j,X[i,j]] + 1
      C_prototypes[cluster,j]=HighestFreq(FrequencyInCluster,clus
      ter,j)
   ENDFOR
ENDFOR
```

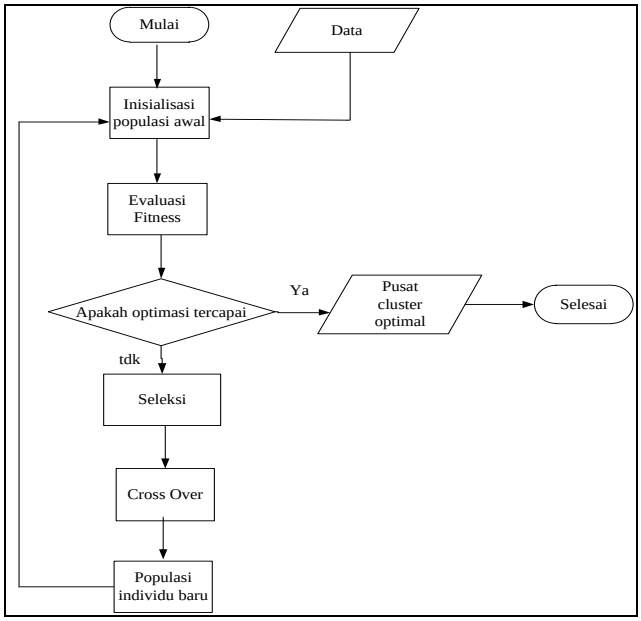
Gambar 3. *Pseudocode* K-Prototype pada tahap pengalokasian objek kedalam cluster (Huang 1998)

Lampiran 4

```
moves=0
FOR i = 1 TO NumberOfObjects
  …
  (To find the cluster whose prototype is the nearest to object
  i.)
  …
   IF (Clustership[i]<>cluster)
    moves+1
    oldcluster=Clustership[i]
    ClusterCount[cluster] + 1
    ClusterCount[oldcluster] - 1
      FOR j=1 TO NumberOfNumericAttributes
       SumInCluster[cluster,j] + X[i,j]
       SumInCluster[oldcluster,j] - X[i,j]
       O_prototypes[cluster,j]=SumInCluster[cluster,j]/ClusterCo
       unt[cluster]
       O_prototypes[oldcluster,j]=
       SumInCluster[oldcluster,j]/ClusterCount[oldcluster]
      ENDFOR
      FOR j=1 TO NumberOfCategoricAttributes
       FrequencyInCluster[cluster,j,X[i,j]] + 1
       FrequencyInCluster[oldcluster,j,X[i,j]] - 1
       C_prototypes[cluster,j]=HighestFreq(cluster,j)
       C_prototypes[oldcluster,j]=HighestFreq(oldcluster,j)
      ENDFOR
  ENDIF
ENDFOR
```

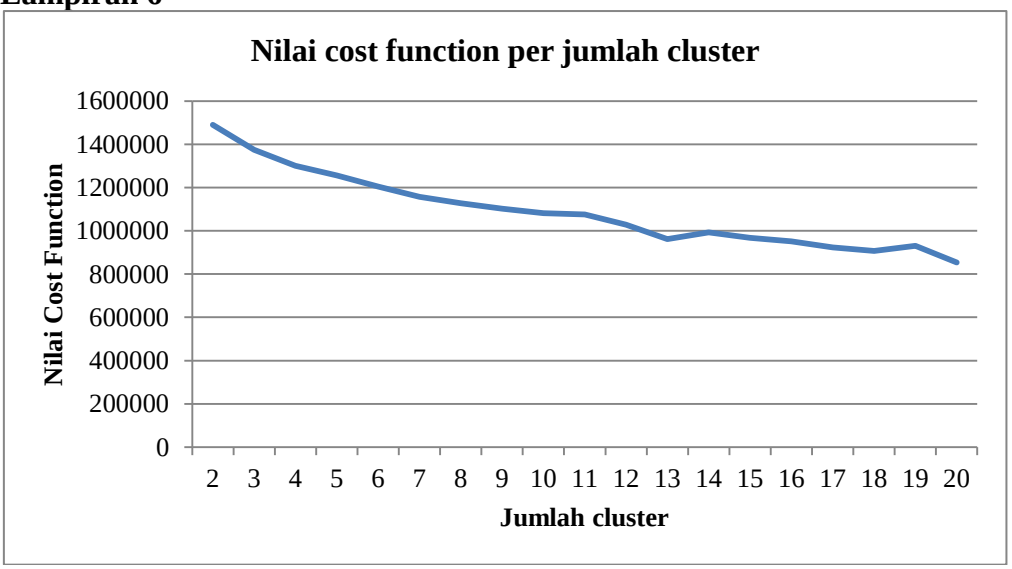
Gambar 4. *Pseudocode* Realokasi Objek

Lampiran 5



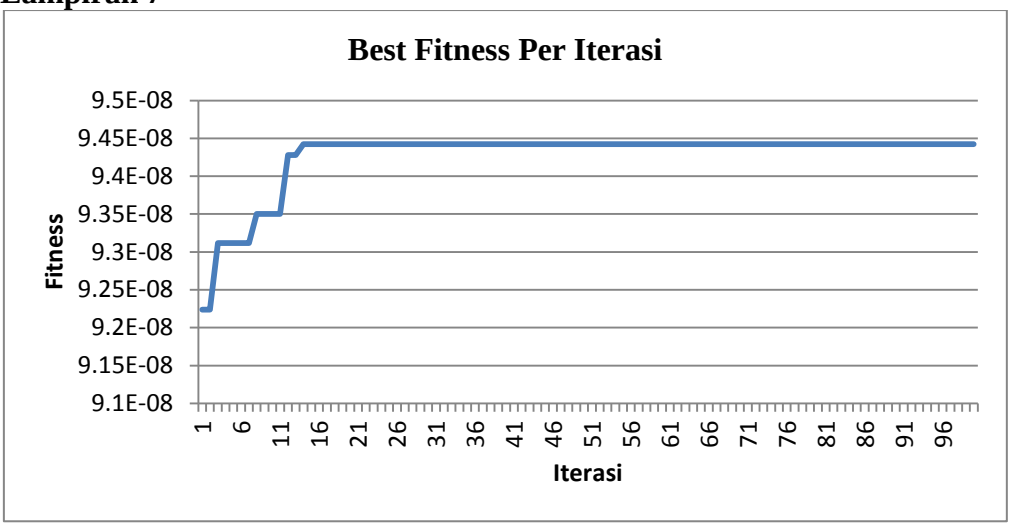
Gambar 5. Alur Proses KPrototype-Genetika untuk center cluster optimal

Lampiran 6



Gambar 6. Perubahan nilai *cost* menurut jumlah *cluster*

Lampiran 7



Gambar 7. Nilai fitness terbaik per iterasi

Lampiran 8

Variabel kategorikal

no objek	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
180	3	1	1	5	5	2	2	2	2	2	2	2	2	2	2	1	4	2	1	2
13158	3	1	1	1	1	2	2	2	2	2	2	2	2	2	2	1	5	2	1	7
26151	3	1	1	1	1	2	2	2	2	2	2	2	2	1	2	1	2	2	1	4
39048	2	1	1	1	1	2	2	2	2	2	2	2	2	2	1	1	4	2	2	2
52138	3	1	2	2	1	2	2	2	2	2	2	2	2	2	2	1	4	2	1	5
65201	3	1	1	2	1	2	2	2	2	2	2	2	2	1	2	1	4	5	1	3

variabel numerik

no objek	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
180	-0.4872	0.2892	-0.5935	-0.1693	0.1555	-0.6512	-0.3237	-0.203	0.6001	-0.2835	-0.0913	-0.125	-0.1352	0.161	-0.0197	-0.3418	-0.1246
13158	0.24763	-0.063	0.13121	-0.1693	0.1463	-0.6512	0.28498	0.4323	-1.116	0.2129	-0.0913	-0.125	-0.1352	-0.047	0.0987	-0.3418	-0.1246
26151	-0.5239	0.2411	-1.0819	-0.1693	0.6412	-0.126	0.45527	0.2744	0.588	-0.2835	-0.0913	-0.125	-0.1352	0.176	-0.5141	-0.3418	-0.1246
39048	-0.2667	-0.64	0.63537	-0.1693	-0.213	-0.6512	-0.0546	0.2465	0.4948	-0.2835	-0.0913	-0.125	-0.1352	-0.047	-0.5141	-0.3418	-0.1246
52138	-0.4504	-0.095	-0.373	-0.1693	-0.379	-0.6512	-0.5677	1.0667	0.6001	-0.2835	-0.0913	-0.125	-0.1352	-0.428	-0.5141	-0.3418	-0.1246
65201	-0.4504	0.6896	-0.5148	-0.1693	-0.234	-0.6512	-0.4333	0.7323	-0.008	-0.2835	-0.0913	-0.125	-0.1352	-0.268	0.1207	0.3074	-0.1246

Gambar 8. Atribut pada objek desa terpilih sebagai inisial center cluster