JURNAL INSY PRO Information System and Processing
Volume 8, Nomor 1, Mei 2023 P –ISSN 2597-419X, E-ISSN 2579-468X
Ojs : http://journal.uinalauddin.ac.id/index.php/insypro
Email : insypro@uin-alauddin.ac.id

ANALISIS KLASTER BERBASIS KEPADATAN DENGAN DBSCAN DAN OPTICS

DENSITY-BASED CLUSTERING ANALYSIS WITH DBSCAN AND OPTICS

Mustikasari[1)], Nur Salman[2)]

[1]Fakultas Sains dan Teknologi Universitas Islam Negeri Makassar
[1]Jl. H. M. Yasin Limpo No.36 Kel. Romang Polong, Kec. Somba Opu, Kab. Gowa, Kode Pos, 92118
[2]Program Studi Teknik Informatika Universitas Dipa Makassar
[2]Jl. Perintis Kemerdekaan KM.9 Tamalanrea Indah, Kec. Tamalanrea, Kota Makassar, Kode Pos 90245

E-mail: mustikasari@uin-alauddin.ac.id[1)], nursalman.halim@undipa.ac.id[2)]

Abstrak – Naskah ini memaparkan proses analisis klaster pada algoritma pengelompokan berbasis kepadatan DBSCAN dan algoritma augmentasi OPTICS yang di implementasi di R. Dibandingkan dengan implementasi lain, DBSCAN menawarkan implementasi yang dapat memanfaatkan data tingkat lanjut seperti pohon k-d untuk mempercepat perhitungan. Keuntungan penting dari implementasi ini adalah kemampuan kedua algoritma dalam menangani dataset besar, khususnya data *granular* dengan berbagai bentuk, yang seringkali tidak dapat ditangani oleh algoritma klaster konvensional berbasis partisi karena sulitnya mengidentifikasi pusat klaster data. Perbandingan sederhana ditunjukkan untuk memberi makna mendalam atas keunggulan metode berbasis kepadatan ini. Eksperimen dengan implementasi terhadap DBSCAN dan OPTICS dibandingkan dengan algoritma populer lainnya menunjukkan bahwa DBSCAN yang di implementasi di R memberikan solusi yang cepat, tangguh dan efisien.

Kata Kunci: analisis, pengelompokan, kepadatan

Abstract *– This paper describes the process of cluster analysis on the DBSCAN density-based clustering algorithm and the OPTICS augmentation algorithm implemented in R.. Compared to other implementations, DBSCAN offers an implementation that can leverage advanced data such as k-d trees to speed up calculations. An important advantage of this implementation is the ability of both algorithms to handle data, especially granular data with various forms, which conventional distance-based separation algorithms often cannot handle because of the difficulty of identifying the center of a data klaster. A simple comparison is shown to give insight into the advantages of this density-based method. Experiments with the implementation of DBSCAN and OPTICS compared to other popular algorithms show that DBSCAN implemented in R provides an fast, strong and efficient solution.*

Keywords: *analysis, clustering, density*

PENDAHULUAN

Penambangan data adalah proses penggalian tersembunyi atas pola atau karakteristik yang menarik dari data dan menggunakannya dalam pengambilan keputusan dan prediksi perilaku masa depan. Hal ini meningkatkan kebutuhan untuk efisien dan metode analisis yang efektif untuk memanfaatkan informasi ini. Salah satu tugas ini adalah pengelompokan tanpa pengawasan yang dikenal dengan istilah "*clustering*" atau analisis klaster. *Clustering* biasanya digambarkan sebagai proses menemukan struktur dalam data dengan mengelompokkan objek yang serupa bersama-sama, di mana kumpulan grup yang dihasilkan disebut klaster. Banyak algoritma pengelompokan secara langsung menerapkan gagasan bahwa kelompok dapat dibentuk sedemikian rupa sehingga objek dalam kelompok yang sama harus lebih mirip satu sama lain daripada objek di kelompok lain. Gagasan kesamaan (atau jarak) berasal dari fakta bahwa objek di asumsikan titik data tertanam dalam ruang data di mana ukuran kesamaan dapat didefinisikan. Kerugian di sebagian besar pengelompokan algoritma tradisional adalah kompleksitas komputasi yang tinggi dan metodenya tidak berskala dengan baik dengan ukuran dataset yang besar. Oleh karena itu, pengembangan algoritma

klastering terus menerima banyak perhatian (penelitian) dalam beberapa dekade terakhir.

Analisis klaster adalah langkah awal dan mendasar dalam analisis data. Ini adalah klasifikasi pola ke dalam kelompok-kelompok data yang tidak diawasi. Secara intuitif, pola dalam klaster yang valid lebih mirip satu sama lain dan berbeda jika dibandingkan dengan pola milik klaster lain. Teknik ini berguna dalam beberapa bidang seperti analisis pola, pembelajaran mesin serta banyak bidang lainnya.

Analisis Klaster dapat diklasifikasikan ke dalam lima jenis utama: metode Partisi, metode Hirarki, berbasis Densitas, berbasis Kisi, dan berbasis Model.

Algoritma partisi bekerja untuk menentukan k partisi yang mengoptimalkan fungsi tujuan tertentu di mana setiap objek ditugaskan ke klaster terdekat. klaster biasanya adalah diwakili oleh rata-rata data seperti algoritma k-means atau dengan objek yang paling terpusat seperti algoritma k-medoid.

Metode partisi (k-means, pengelompokan PAM) dan pengelompokan hierarki bekerja dengan baik untuk klaster yang kompak dan terpisah dengan baik. Di sisi lain, metode partisi dapat kesulitan dalam menemukan pusat titik data pada data *granular* yang padat. Selain itu, metode partisi juga sangat terpengaruh oleh adanya *derau* dan *outlier* dalam data. Tantangan *real-world* data dapat berupa:

i) kelompok bentuk sembarang (kelompok bentuk oval, linier, menyerupai huruf "S", dan bentuk pola acak);

ii) banyak *outlier* dan *derau*.

Pengelompokan berbasis kepadatan juga dapat menangani derau, di mana titik-titik yang berada di area dengan kepadatan sangat rendah tidak diberi label klaster, melainkan diperlakukan sebagai *outlier* atau pengamatan derau. Properti ini memberikan keuntungan untuk banyak aplikasi. Misalnya, kumpulan data medis mungkin penuh dengan titik data berisik atau "*derau*", karena kesalahan estimasi saat data diambil atau kendala dalam proses penyaringan atas sumber data, di mana kendala bentuk fisik yang diasumsikan oleh metode berbasis model lebih mungkin untuk dilanggar.

Algoritma berbasis kepadatan tidak memerlukan jumlah klaster terlebih dahulu karena secara otomatis mendeteksi klaster dengan nomor aslinya. Salah satu algoritma berbasis kepadatan adalah DBSCAN. Kelemahan dasar dari teknik ini adalah kompleksitas komputasi yang tinggi, dimana metode ini menjelajahi semua tetangga dalam memeriksa kondisi inti untuk setiap objek khususnya ketika algoritma bekerja pada dataset yang sangat besar, langkah ini akan sangat mahal.

Artikel ini menyajikan ikhtisar paket R dbscan yang berfokus pada DBSCAN dan OPTIK, menguraikan operasinya dan secara eksperimental membandingkan kinerjanya dengan perubahan nilai parameter dalam beberapa percobaan.

Isi lengkap naskah ini selanjutnya disusun sebagai berikut: Diskusi lebih lanjut tentang algoritma DBSCAN dan OPTICS diberikan di bagian 2. Bagian 3 menyajikan algoritma yang di implementasi. Bagian 4 membahas hasil eksperimen dan pengaruh parameter yang berbeda pada kinerja keseluruhan. Terakhir adalah pembahasan dan kesimpulan yang disajikan pada bagian 5.

METODOLOGI PENELITIAN

1. Paket dan Cara Kerja Algoritma DBSCAN

DBSCAN (*Density Based Spatial Clustering of Applications with Noise*) atau pengelompokan spasial berbasis kepadatan pada aplikasi dengan derau adalah algoritma berbasis kepadatan yang dapat mendeteksi klaster berbentuk berbeda-beda. DBSCAN biasanya menganggap klaster sebagai daerah padat objek dalam ruang data yang dipisahkan oleh kepadatan daerah rendah. DBSCAN adalah teknik pengelompokan berbasis kepadatan yang dimulai dari objek mana saja dan jika tetangga di sekitarnya dalam radius tertentu (ε) memenuhi setidaknya jumlah minimum objek (minPts). Objek ini merupakan objek inti dan pencarian secara rekursif berlanjut dengan tetangganya dan akan berhenti di perbatasan objek di mana semua titik dalam klaster harus berada tetangga salah satu objek intinya. Objek "*ungroup*" (belum berkelompok) mana saja dalam dataset ditempatkan di dalam klaster. Semua objek non-inti yang tidak berada di tetangga salah satu objek inti diberi label sebagai *derau*. DBSCAN tidak membutuhkan di awal proses, angka jumlah klaster akhir, karena algoritma ini secara otomatis mendeteksi daerah padat dan hasilnya adalah jumlah klaster alami. DBSCAN sebagai algoritma berbasis kepadatan, dapat mendeteksi klaster berbentuk beragam. Algoritma ini mencari objek inti untuk mengekstrak daerah padat dan akhirnya merupakan bagian dari klaster. Memeriksa setiap objek untuk kondisi inti, artinya mencari seluruh dataset untuk mendapatkan tetangga dalam ambang ε yang ditentukan. Algoritma DBSCAN di cantumkan dalam *pseudo code* pada Gambar 1

JURNAL INSY PRO Information System and Processing
Volume 8, Nomor 1, Mei 2023 P –ISSN 2597-419X, E-ISSN 2579-468X
Ojs : http://journal.uinalauddin.ac.id/index.php/insypro
Email : insypro@uin-alauddin.ac.id

Algoritma: : DBSCAN: algoritma klaster berbasis Densitas.
Input: D: Kumpulan data da yang berisi ***n*** *objek*;
∈ :parameter radius,
MinPte: ambang Densitas lingkungan;
Output: Satu set klaster berbasis Densitas ;
Metode:

1. tandai semua objek sebagai belum dikunjungi;
2. **do**
   secara acak memilih objek yang belum dikunjungi ***p*** beri tanda begitu dikunjungi:
3. **if** ∈-neighborhood ***p*** memiliki setidaknya objek *MinPts*,
   buat klaster baru *C*, dan tambahkan ***p*** ke *C*
   tetapkan *N* menjadi himpunan objek di ∈-neighborhood ***p***
4. **for each** titik ***p′*** di ***N***;
5. **if** ***p*** tidak dikunjungi
   tandai ***p′*** sebagai dikunjungi:
6. **if** di neighborhood dari ***p′*** memiliki setidaknya poin *MinPts*
   tambahkan titik-titik itu ke ***N***;
7. **if** ***p′*** belum menjadi anggota klaster mana pun, tambahkan ***p′*** ke C
8. **End for**
   output **C**
9. **else** tandai p sebagai kebisingan;
10. **until** tidak ada objek yang belum dikunjungi;

Gambar 1. Algoritma DBSCAN

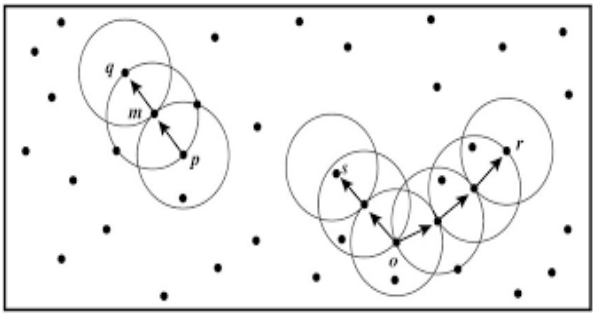
Dalam pengelompokan berbasis kepadatan, daerah padat dalam ruang data dipisahkan dari daerah dengan kepadatan lebih rendah. Pengamatan ditugaskan ke klaster tertentu jika kepadatan di lokasi tertentu lebih besar dari ambang batas yang telah ditentukan. Untuk pengamatan yang diberikan dalam satu klaster, kepadatan lokal di sekitar titik tersebut harus melebihi ambang batas tertentu. Kepadatan lokal ditentukan oleh dua parameter: radius ε lingkaran yang berisi sejumlah tetangga di sekitar titik tertentu dan jumlah minimum titik di sekitar radius tersebut: minPts.

Terdapat lebih dari minPts titik di ε-*neighborhood*, yaitu:

1. Titik batas: Lebih sedikit dari minPts dalam ε tetapi di sekitar titik inti.

2. Titik *derau* atau *Outlier*: Semua titik yang tersisa: Bukan titik inti, dan tidak cukup dekat untuk dapat dijangkau dari titik inti.

Ini dimulai dengan memilih secara acak sebuah titik yang belum ditetapkan ke sebuah klaster. Kemudian algoritma menentukan apakah itu adalah titik inti atau *outlier*. Setelah menemukan titik inti, semua pengamatan yang dapat dijangkau kepadatannya akan ditambahkan ke sebuah klaster. Setelah itu, Algoritma akan melakukan lompatan tetangga ke setiap titik yang dapat dijangkau secara langsung dan menambahkannya ke klaster. Jika *outlier* telah ditambahkan, itu akan diberi label sebagai titik batas. Langkah algoritma kemudian mengambil titik inti lain dan mengulangi langkah sebelumnya sampai semua titik telah ditetapkan ke klaster atau diberi label sebagai *outlier*.



Gambar 2. Densitas-jangkauan dan Densitas-konektivitas dalam klaster berbasis Densitas. (Sumber: Data Mining Concepts and Tecniques, Second edition, p419)

Karena merupakan metode berbasis kepadatan, DBSCAN tidak memerlukan seseorang untuk menentukan jumlah klaster dalam data secara apriori, berbeda dengan kebanyakan skema partisi. Selain itu, DBSCAN dapat menemukan klaster berbentuk sembarang. Namun, pemilihan kombinasi yang tepat dari parameter minPts dan ε dapat menjadi sulit ketika densitas di seluruh konfigurasi tidak seragam atau ketika terdapat struktur hierarki. Untuk mengatasi masalah ini, salah satu dari beberapa varian telah diusulkan, dan relevansi khususnya adalah OPTICS yang memberikan pandangan hirarkis dari struktur klaster.

2. Paket dan Cara Kerja Algoritma OPTICS

OPTICS meminjam konsep inti yang dapat dicapai dengan densitas dari DBSCAN. Tetapi sementara DBSCAN dapat dianggap sebagai algoritma pengelompokan, mencari kelompok alami dalam data, OPTICS adalah algoritma pengurutan tambahan yang darinya hasil pengelompokan datar atau hierarkis dapat diturunkan. OPTICS membutuhkan parameter ε dan minPts yang sama dengan DBSCAN, bagaimanapun, parameter ε secara teoritis tidak diperlukan dan hanya digunakan untuk tujuan praktis mengurangi kompleksitas *runtime* dari algoritma. Untuk mendeskripsikan OPTICS, kami memperkenalkan konsep tambahan yang disebut jarak inti dan jarak jangkauan. Semua jarak yang digunakan dihitung menggunakan metrik yang sama (jarak Euclidean adalah yang paling sering dipilih) yang digunakan untuk perhitungan tetangga.

Ada dua cara mengekstraksi klaster, baik melalui global ε′ atau ambang ξ relatif, OPTICS dapat dilihat sebagai generalisasi DBSCAN. Dalam konteks di mana seseorang ingin menemukan klaster dengan kepadatan yang sama, ExtractDBSCAN OPTICS menghasilkan solusi seperti DBSCAN, sementara dalam konteks lain Extract-ξ dapat menghasilkan hierarki yang mewakili klaster dengan kepadatan yang

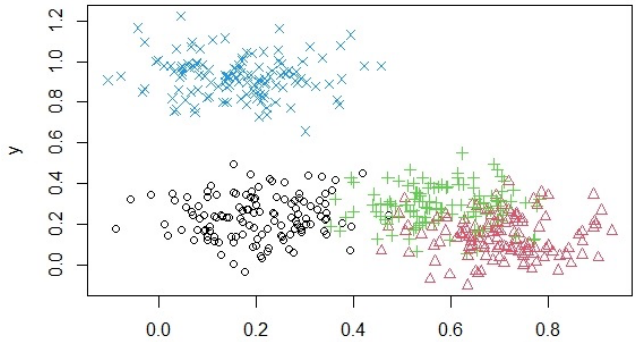
bervariasi. Oleh karena itu menarik untuk dicatat bahwa sementara DBSCAN telah mencapai pujian kritis, bahkan memotivasi banyak pengembangan, OPTICS jelas kurang mendapat perhatian. Mungkin salah satu alasannya adalah karena metode Extract-ξ untuk mengelompokkan poin ke dalam klaster sebagian besar tidak diperhatikan, dan karenanya tidak diterapkan di sebagian besar paket perangkat lunak sumber yang mengiklankan implementasi OPTICS.
Faktanya, mungkin karena implementasi Extract-ξ kluster OPTICS (tidak lengkap) di berbagai pustaka perangkat lunak, ada beberapa kebingungan mengenai penggunaan OPTICS, dan manfaat yang ditawarkannya dibandingkan dengan DBSCAN. Beberapa makalah memotivasi pengembangan DBSCAN atau menyusun algoritma baru dengan mengutip metode OPTICS sebagai algoritma yang tidak mampu menemukan kluster densitas-heterogen, seperti yang dikutip dalam (Ghanbarpour dan Minaei, 2014). Namun, OPTICS pada dasarnya mengembalikan urutan data yang dapat diproses lebih lanjut untuk mengekstraksi, yaitu 1) pengelompokan datar pada kluster dengan kepadatan yang relatif sama atau 2) hierarki kluster, yang adaptif untuk merepresentasikan kepadatan lokal dalam data.

Algoritma berbasis kepadatan di DBSCAN sangat bergantung pada pembentukan tetangga, yaitu, menemukan semua titik milik tetangga ε. Pendekatan sederhana adalah melakukan pencarian linier, yaitu selalu menghitung jarak ke semua titik lain untuk menemukan titik terdekat. Ini membutuhkan $O(n)$ operasi, dengan n adalah jumlah titik data, untuk setiap kali suatu tetangga diperlukan. Karena DBSCAN dan OPTICS memproses setiap titik data satu kali, ini menghasilkan kompleksitas *runtime* $O(n^2)$. Cara mudah dalam R adalah menghitung matriks jarak dengan semua jarak berpasangan antara titik dan mengurutkan jarak untuk setiap titik (baris dalam matriks jarak) untuk menghitung terlebih dahulu tetangga terdekat untuk setiap titik. Namun, metode ini memiliki kekurangan yaitu ukuran matriks jarak penuh adalah $O(n^2)$, dan menjadi sangat besar dan lambat untuk menghitung dataset sedang hingga besar. Untuk menghindari penghitungan matriks jarak lengkap, DBSCAN sendiri bergantung pada struktur data partisi-ruang yang disebut pohon k-d. Struktur data ini memungkinkan DBSCAN untuk mengidentifikasi kNN atau semua tetangga dalam radius tetap ε yang lebih efisien dalam waktu sub-linear dengan menggunakan rata-rata hanya $O(\log(n))$ operasi per-*query*. Ini mampu menghasilkan pengurangan nilai kompleksitas *runtime* menjadi $O(n\log(n))$. Namun, perlu diperhatikan bahwa pohon k-d diketahui mengalami degenerasi untuk data berdimensi tinggi yang membutuhkan operasi $O(n)$ dan menghasilkan kinerja yang tidak lebih baik daripada pencarian linear. Pencarian kNN cepat dan pencarian tetangga terdekat dengan radius tetap ini, sama-sama telah di adopsi dalam DBSCAN dan OPTICS di R.

HASIL DAN PEMBAHASAN

Kami menggunakan set data buatan dari empat distribusi *Gaussian* dengan sebaran data yang sedikit tumpang tindih dalam ruang dua dimensi dengan masing-masing distribusi terdiri atas 125 titik-titik data. Kami memuat DBSCAN, mengatur generator angka acak agar hasilnya dapat direproduksi dan membuat data set.

Data set yang dihasilkan tersebut ditunjukkan pada Gambar 3.



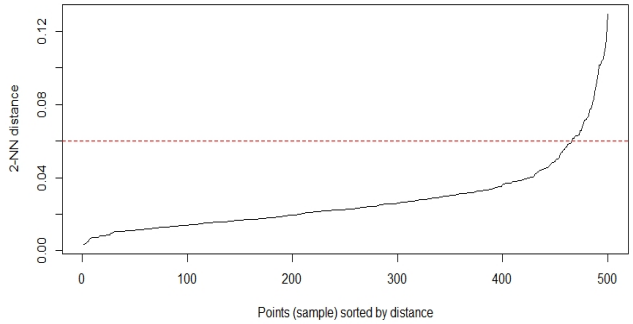
Gambar 3: Data set sampel, terdiri dari 4 distribusi *Gaussian*

Untuk menerapkan DBSCAN, kita perlu memutuskan ε radius tetangga dan minPts ambang kepadatan. Aturan praktis untuk minPts adalah menggunakan setidaknya jumlah dimensi kumpulan data ditambah satu. Dalam kasus kita, ini adalah 3. Untuk ε, kita dapat memplot jarak titik kNN (yaitu, jarak ke tetangga terdekat ke-k) dalam urutan menurun dan mencari lengkungan di plot data. Ide di balik heuristik ini adalah bahwa titik-titik yang terletak di dalam klaster akan memiliki jarak k-tetangga terdekat yang kecil, karena mereka dekat dengan titik lain dalam klaster yang sama, sedangkan titik *derau* di isolasi dan akan memiliki jarak kNN yang agak besar. DBSCAN menyediakan fungsi yang disebut kNNdistplot() di R untuk membuatnya lebih mudah. Untuk k kami menggunakan minPts-1 karena minPts DBSCAN menyertakan titik data aktual dan jarak ke-k tetangga terdekat tidak tercakup. Untuk menerapkan DBSCAN, kita perlu memutuskan ε radius tetangga

Volume 8, Nomor 1, Mei 2023 P –ISSN 2597-419X, E-ISSN 2579-468X
Ojs : http://journal.uinalauddin.ac.id/index.php/insypro
Email : insypro@uin-alauddin.ac.id

dan ambang batas kepadatan. Untuk ε, kami dapat memplot jarak titik-titik (mis., Jarak ke tetangga terdekat ke k) dalam urutan menurun dan mencari lengkung di plot. Gagasan di balik heuristik ini adalah bahwa titik-titik yang terletak di dalam klaster akan memiliki jarak tetangga *k-Nearest* yang kecil, karena mereka dekat dengan titik-titik lain di klaster yang sama, sementara titik-titik derau diisolasi dan akan memiliki jarak kNN yang agak besar. DBSCAN menyediakan fungsi yang disebut kNNdistplot() untuk mempermudah ini.

Untuk k kami menggunakan minPts-1 karena minPts DBSCAN mencakup titik data aktual dan jarak ke k tetangga terdekat tidak tercakup.

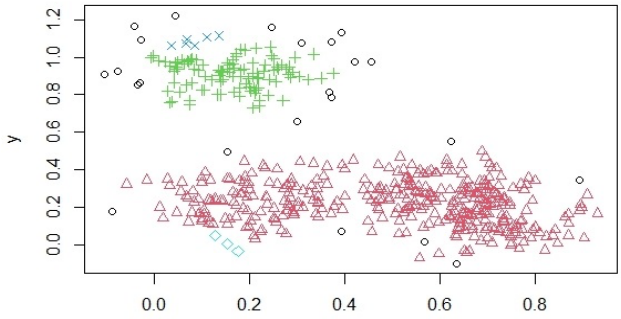


Gambar 4: Plot Data k-NN Distance

Selanjutnya, jarak kNN ditampilkan sebagaimana yang ditunjukkan pada Gambar 4. lengkung terlihat di sekitar jarak 2-nn 0,06. Ada tambahan garis horizontal yang menandai untuk referensi. Kemudian, kita dapat melakukan pengelompokan dengan parameter yang dipilih.

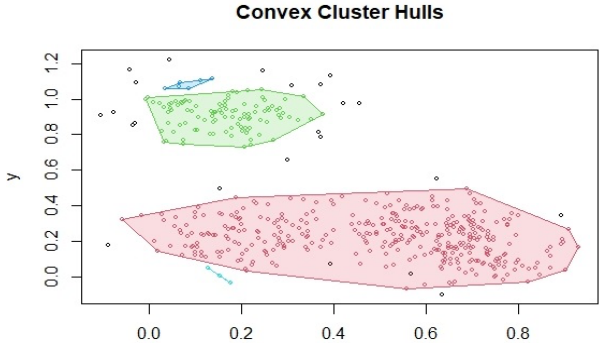
```
DBSCAN klastering for 500 objects.
Parameters: ϵ = 0.06, minPts = 3
The klastering contains 4 klaster(s) and 23
derau points.

   0   1   2   3   4
  23 365 103   6   3
Available fields: klaster, ϵ, minPts
```

Proses klasterisasi data yang dihasilkan mengidentifikasi satu klaster besar dengan 365 titik anggota, dua kelompok menengah dengan sekitar 103 titik, dan dua kelompok yang sangat kecil dengan total 9 titik serta 23 titik derau (diwakili oleh klaster pertama yaitu klaster-ID 0).



Gambar 5: Hasil klasterisasi dengan DBSCAN. (derau direpresentasikan sebagai lingkaran hitam)

Plot pencar pada Gambar 5 menunjukkan bahwa algoritma pengelompokan dengan benar mengidentifikasi dua klaster atas, tetapi menggabungkan dua klaster yang lebih rendah karena wilayah di antara mereka memiliki kepadatan yang cukup tinggi. klaster kecil adalah kelompok terisolasi dari 3 titik (lewat minPts) dan titik-titik derau terisolasi titik. klaster kecil ini dapat ditekan dengan menggunakan sejumlah besar minPts. DBSCAN juga menyediakan plot yang menambahkan klaster Convex-Hulls ke plot pencar yang ditunjukkan pada Gambar 6.



Gambar 6: Plot Convex-Hulls dari klasterisasi DBSCAN, derau berwarna hitam.

Analisis klaster juga dapat digunakan untuk mencari tahu bahwa klaster titik data baru akan ditetapkan menggunakan prediksi. Metode prediksi menggunakan penugasan tetangga terdekat ke titik inti dan membutuhkan dataset asli. Parameter tambahan diteruskan ke metode pencarian tetangga terdekat. Di sini kami mendapatkan penugasan klaster untuk 25 titik data pertama. Perhatikan bahwa penugasan ke klaster 0 berarti bahwa titik data dianggap derau karena tidak cukup dekat dengan titik inti.

```
> predict(res, x[1:25,], data = x)
 [1] 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1
2 1 1 1 0
```

Kecuali OPTICS murni digunakan untuk mengextract pengelompokan DBSCAN, parameternya memiliki efek yang berbeda untuk DBSCAN: ε

Volume 8, Nomor 1, Mei 2023 P –ISSN 2597-419X, E-ISSN 2579-468X
Ojs : http://journal.uinalauddin.ac.id/index.php/insypro
Email : insypro@uin-alauddin.ac.id

biasanya dipilih agak besar (kami menggunakan 10 di sini) dan minPts sebagian besar mempengaruhi perhitungan inti dan jangkauan jarak, di mana nilai yang lebih besar memiliki Efek *smoothing*. Dengan menggunakan minPts=10, yaitu, jarak inti didefinisikan sebagai jarak ke tetangga terdekat ke 9 (mencakup tetangga 10 titik).

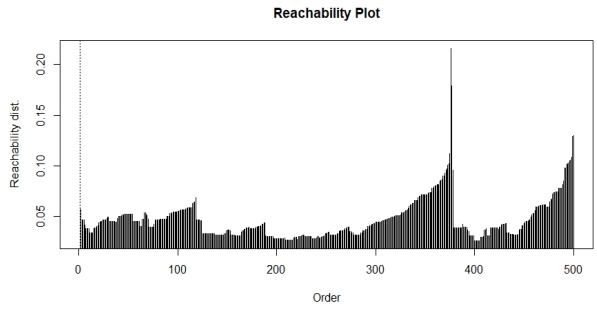
```
OPTICS ordering/klastering for 500 objects.
Parameters: minPts = 10, ϵ = 10, ϵ_cl = NA,
xi = NA
Available fields: order, reachdist,
coredist, predecessor,
minPts, ϵ, ϵ_cl, xi
> head(res$order, n = 15)
 [1]   1 453 363 293 285 325 457 329 301
257 241 169 153  77  17
```

OPTICS adalah algoritma pemesanan *augmented*, yang menyimpan urutan yang dihitung dari titik -titik yang ditemukan dalam elemen urutan objek yang dikembalikan.

Ini berarti bahwa titik data 1 dalam set data adalah yang pertama dalam urutan, titik data 453 adalah yang kedua, dan seterusnya. Urutan berbasis kepadatan yang dihasilkan oleh OPTICS dapat langsung diplot sebagai plot jangkauan.

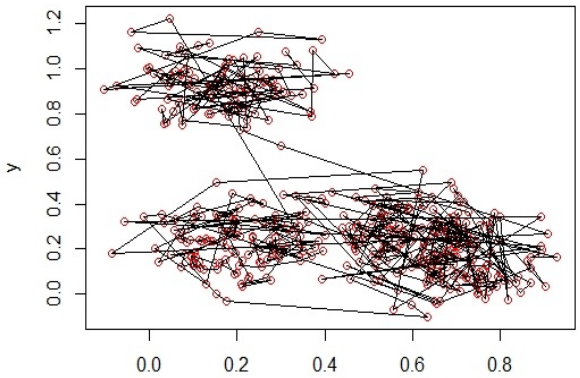
```
> plot (res)
```

Plot jangkauan pada Gambar 7 menunjukkan jarak jangkauan untuk titik yang dipesan oleh OPTICS. Lembah mewakili klaster potensial yang dipisahkan oleh puncak. Puncak yang sangat tinggi dapat menunjukkan titik derau. Untuk memvisualisasikan urutan set data asli, kami dapat memplot garis yang menghubungkan titik-titik tersebut.



Gambar 7 Contoh plot jangkauan optik untuk set data dengan empat kelompok masing -masing 500 titik data.

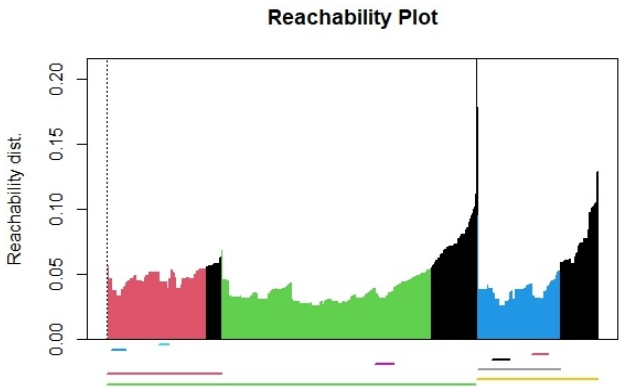
titik-titik di setiap klaster dikunjungi secara berurutan dimulai dengan titik-titik di tengah (wilayah terpadat) dan kemudian titik -titik di daerah sekitarnya.



Gambar 8: Urutan titik data OPTIK direpresentasikan sebagai garis.

Gambar 8 menunjukkan bahwa titik-titik pada setiap kluster dikunjungi secara berurutan dimulai dari titik-titik di tengah (wilayah terpadat) kemudian titik-titik di wilayah sekitarnya.

OPTICS memiliki dua metode ekstraksi klaster utama menggunakan struktur keterjangkauan terurut yang dihasilkannya. Pengelompokan tipe DBSCAN dapat diekstraksi menggunakan *extract*DBSCAN() dengan menentukan parameter ε global. Plot *reachability* pada gambar 7 menunjukkan empat puncak yaitu titik-titik dengan jarak *reachability* yang tinggi. Titik-titik tersebut menunjukkan batas antara klaster empat klaster. Ambang ε yang memisahkan keempat kluster dapat ditentukan secara visual. Dalam hal ini, kami menggunakan ε_cl dari 0.055. Jangkauan yang dihasilkan dan kluster yang sebagaimana yang ditunjukkan Gambar 9 dan 10.

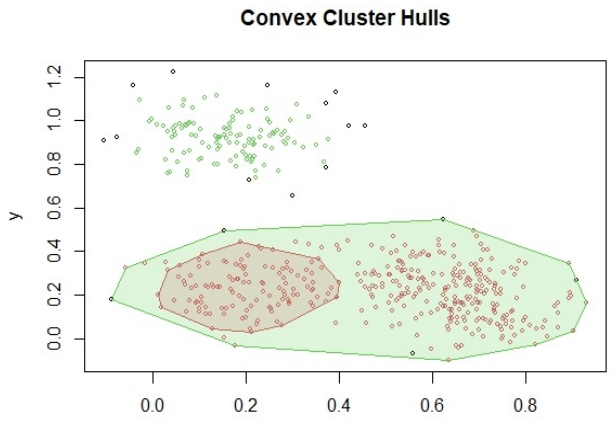


Gambar 9: Plot jangkauan untuk pengelompokan tipe DBSCAN yang di extract-ξ di global ε = 0,055 hasil dalam empat kluster.

Pengelompokan menyerupai struktur asli dari empat kelompok yang dihasilkan data, dengan satu-satunya perbedaan adalah bahwa titik pada batas kelompok ditandai sebagai titik derau. DBSCAN juga menyediakan ExtractXi() untuk mengekstraksi struktur

Volume 8, Nomor 1, Mei 2023 P –ISSN 2597-419X, E-ISSN 2579-468X
Ojs : http://journal.uinalauddin.ac.id/index.php/insypro
Email : insypro@uin-alauddin.ac.id

kluster hierarkis. Kami menggunakan nilai XI=0.055 disini.



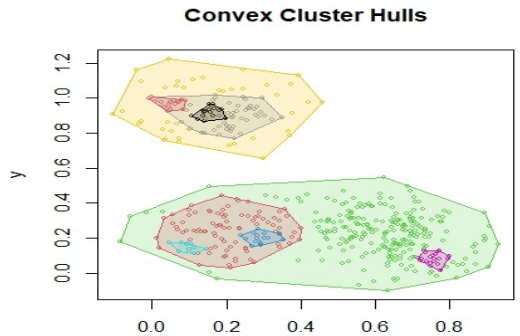
Gambar 10: Plot "*Convex-hulls*" untuk pengelompokan tipe DBSCAN yang diextractsi di global ε = 0,055 hasil dalam empat kluster.

Metode ξ menghasilkan struktur pengelompokan hierarkis, dan dengan demikian titik dapat menjadi anggota dari beberapa kelompok bersarang.

Klaster direpresentasikan sebagai rentang yang berdekatan dalam plot jangkauan dan tersedia di bidang klasters_xi. Di sini kita memiliki tujuh kelompok. klaster juga terlihat dalam plot jangkauan.

```
> plot (res)
> hullplot (x, res)
```

Gambar 10 menunjukkan plot jangkauan dengan klaster yang direpresentasikan menggunakan warna dan batang vertikal di bawah plot. klaster tersebut juga dapat diplot dengan fungsi plot "*convex-hull*" yang ditunjukkan pada Gambar 11. Perhatikan bagaimana struktur bersarang ditunjukkan oleh klaster di dalam kelompok. Perhatikan juga bahwa dimungkinkan untuk "*convex-hull*", sebagai visualisasi, untuk mengandung titik yang tidak dianggap sebagai bagian dari pengelompokan klaster.

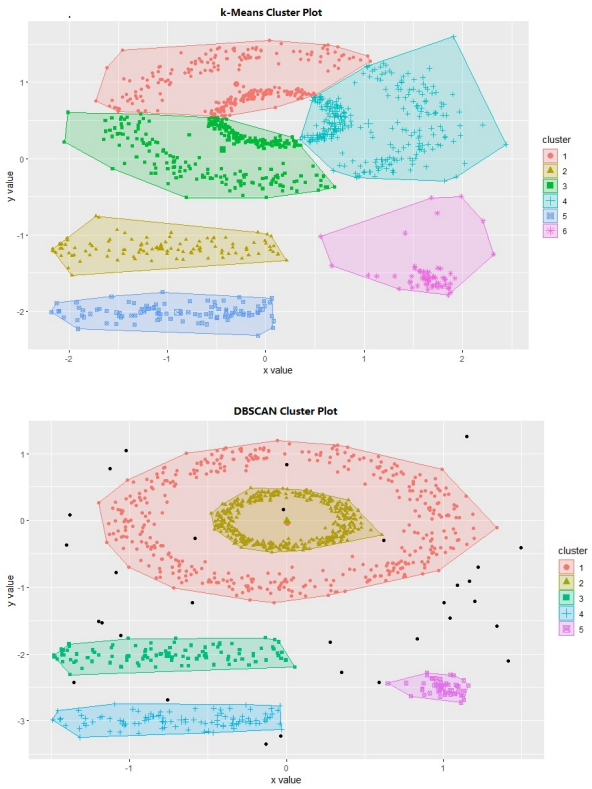


Gambar 11: Plot lambung cembung dari pengelompokan hierarkis yang diextractsi dengan extract-ξ

Di sini kita memiliki tujuh kelompok. Cluster juga terlihat dalam plot jangkauan.

```
OPTICS ordering/clustering for 500 objects.
   Parameters: minPts = 10, eps = 10, eps_cl = NA, xi = 0.05
   The clustering contains 9 cluster(s) and 0 derau points.

   Available fields: order, reachdist, coredist, predecessor,minPts, eps, eps_cl, xi, clusters_xi, cluster

   start end cluster_id
1      1 117          1
2      1 375          2
3      5  20          3
4     54  64          4
5    273 292          5
6    376 500          6
7    378 462          7
8    392 410          8
9    432 449          9
```

Percobaan terakhir kami lakukan uji data untuk deteksi bentuk kepadatan data berbeda lainnya. Dan juga melakukan uji serupa pada algoritma partisi, yaitu k-Means yang populer cukup handal menangani data *globular*. Hasilnya kami tunjukkan pada Gambar 12.



Gambar 12. Plot klasterisasi shape data dengan algoritma k-Means(atas) dan DBSCAN (bawah)

Dengan demikian kami telah mengevaluasi kinerja implementasi DBSCAN dan OPTICS DBSCAN pada data set berbagai ukuran dan bentuk. Ini bukan studi evaluasi yang komprehensif, tetapi contoh ini digunakan untuk menunjukkan kinerja implementasi DBSCAN dan OPTICS DBSCAN pada kumpulan data dengan berbagai ukuran dan bentuk.

KESIMPULAN

Hasil eksperimen dan analisis klaster yang di implementasi pada penelitian ini, dapat dirangkum sebagai berikut:

1. Algoritma yang diusulkan dapat digunakan secara efisien untuk mengelompokkan kumpulan data besar dengan berbagai bentuk.
2. Jumlah maksimum tetangga yang disarankan untuk data 2D adalah 3 seperti yang terlihat pada hasil percobaan.
3. Lebih tangguh mengenali data globular dibandingkan teknik partisi seperti k-Means
4. Algoritma DBSCAN dan OPTICS terbukti cukup tangguh mengelompokkan data dengan berbagai ukuran dan bentuk secara cepat tangguh dan efisien di R.

DAFTAR PUSTAKA

Hahsler, M., Piekenbrock, M., & Doran, D. (2019). DBSCAN: Fast Density-Based klastering with R. Journal of Statistical Software, 91(1),1–30. https://doi.org/10.18637/jss.v091.i01

Han, J., Pei, J., & Tong, H. (2022). Data mining: concepts and techniques. Morgan kaufmann.

Hennig, C. (2015). What are the true klasters?. Pattern Recognition Letters, 64, 53-62. https://doi.org/10.1016/j.patrec.2015.04.009

Kassambara, A. (2018). Machine learning essentials: Practical guide in R. Sthda.

Khan, K., Rehman, S. U., Aziz, K., Fong, S., & Sarasvady, S. (2014, February). DBSCAN: Past, present and future. In The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014) (pp. 232-238). IEEE.

Ghanbarpour, A., & Minaei, B. (2014, February). EXDBSCAN: An extension of DBSCAN to detect clusters in multi-density datasets. In 2014 Iranian Conference on Intelligent Systems (ICIS) (pp. 1-5). IEEE.

Scrucca L, Fop M, Murphy TB, Raftery AE (2016). "mclust 5: klastering, Classification and Density Estimation Using Gaussian Finite Mixture Models." The R Journal, 8(1), 205–233. doi:10.32614/RJ-2016-021.

Wierzchoń, S. T., & Kłopotek, M. A. (2015). Algorithms of klaster analysis (Vol. 3). Warszaw, Poland: Institute of Computer Science Polish Academy of Sciences.