

PERBANDINGAN BEBERAPA ALGORITMA  
KLUSTERING UNTUK MENENTUKAN TITIK AWAL  
ITERASI PENCARIAN NILAI MINIMUM FUNGSI

Oleh : Haeni Budiati, S.Si, M.Kom.

Abstract

*Searching minimum values by using iterative methods after is started with a randomly chosen point. From there the iteration process gets closer to the minimum. If the iteration is repeated, it often gets the same minimum.*

*In this research, clustering method is used to group the points, so it can be chosen a representative as first iterate point. Clustering methods that have been tried: PointFirst, PointsBest, MinFirst, MinBest, MaxFirst, MaxBest, AveFirst, and AveBest and the functions have been tried are Rastrigin, Rosenbrock and Goldstein. Random data that has been tried 50 with maximum point distance ( $p$ ) varieties 0,1 - 0,5.*

*Result of the research shows that  $p$  fewer, cluster's amount bigger, clusters formation's time get longer. Point first algorithm is the fastest algorithm. Function that is having many minimum points is suitable by using MaxBest algorithm with big  $p$ . Numerous Newton's iterate relative constant. In general, higher  $\rho$  will get bigger affectivity, the rising up of a value depends on the function has been used.*

*Key words : cluster, iteration, initial point.*

1. PENDAHULUAN.

Dalam perkembangan masalah optimisasi, banyak ditemukan metode-metode yang berfungsi untuk mencari titik optimal. Secara umum pencarian titik optimal dapat digunakan dua cara. Cara pertama cara argumentasi. Cara yang kedua dengan informasi tambahan melalui differensiasi. Selama ini yang sering digunakan untuk mencari harga minimum  $f(x)$  adalah menggunakan differensiasi. Syarat fungsi  $f(x)$  minimum adalah

$$\frac{df}{dx} = 0; \quad \frac{d^2f}{dx^2} > 0.$$

Masalahnya dalam menyelesaikan derivatif pertama sama dengan nol ( $\frac{df}{dx} = 0$ ) belum tentu ada penyelesaian eksak. Untuk itu diperlukan pendekatan numerik. Didalam pendekatan numerik diperlukan titik awal

iterasi yang baik, supaya didapat hasil yang konvergen.

Konsep kluster merupakan suatu hal yang bersifat obyektif dan sulit didefinisikan dengan tepat. Tetapi karena analisis kluster pada dasarnya tergantung pada ukuran kemiripan antara data, maka sebuah kluster dapat didefinisikan sebagai sebuah grup, atau kategori-kategori dari obyek-obyek yang mirip. Apakah obyek itu orang-orang, barang-barang industri, atau titik-titik data abstrak yang mirip dari satu kasus ke kasus.

Misalkan  $x = \{x_1, \dots, x_p\} \in R^p$  adalah sebuah sampel data multivariate diambil pada  $n$  observasi masing-masing dengan  $P$  variabel. Secara teoritis data dapat dikukiskan kedalam ruang berdimensi  $P$ , yang merupakan  $n$  titik dan kluster yang terbentuk didalamnya akan dicari dengan teknik analisis kluster.

Struktur analisis kluster dibagi menjadi dua yaitu Struktur Hierarchi dan Struktur Partisi atau Non Hierarchi.

1. Struktur Hierarchi  
Dalam model ini, sebuah system kluster dikonstruksikan sebagai proses penggabungan kluster-kluster atau proses pemecahan kluster-kluster sebagai hasil akhir akan merupakan sebuah konfigurasi semua kluster yang terdapat didalam sampel data.

2. Struktur Partisi / Non Hierarchi  
Dalam model ini, system kluster diperoleh melalui proses partisi yang dilakukan dengan proses pemilihan dan pemasaan data dari satu kluster ke kluster yang lain, proses ini diteruskan sampai sebuah partisi yang optimal diperoleh.

Secara teoritis teknik analisis kluster non hierarchi memperlakukan satu data ke dalam kluster-kluster melalui urutan partisi seperti didalam teknik analisis kluster hierarchi. Dalam teknik hierarchi proses klasifikasi dilakukan untuk mencari satu bentuk partisi yang baik, yaitu bentuk kluster terpisah dengan baik kedalam  $k$  kluster.

Secara umum proses dari sebuah teknik non hierarchi adalah :

1. Partisi awal data ke dalam kluster-kluster sembarang yang dibentuk.
2. Data didalam setiap kluster diubah-ubah dengan proses mentransfer data satu-satu dari satu kluster ke kluster lain, data dimasukkan ke suatu kluster dengan jarak tertentu terhadap titik-titik maupun klusternya.
3. Proses diulang sampai mencapai suatu bentuk partisi yang optimal, yaitu proses mencapai iterasi yang ditentukan.

Teknik analisis kluster non hierarchi dapat dipergunakan lebih efektif dari pada teknik hierarchi, sebab untuk mengadakan klasifikasi data yang berukuran besar teknik non hierarchi tidak membutuhkan perhitungan matriks disemilarias  $D$  yang membutuhkan memori komputer besar.

Dalam pengecekan data algoritma PointFirst akan menghitung jarak antar semua titik yang kemudian dibandingkan dengan  $\rho$ . Apabila jarak antar titik lebih kecil dari  $\rho$  maka titik tersebut akan menjadi satu kluster dengan titik sebelumnya yang pertama kali ketemu.

Algoritma PointBest dimulai dengan memasukkan data pertama ke kluster mencari jarak data (i) dengan data yang sudah masuk ke kluster, mencari jarak minimal dan dibandingkan dengan  $\rho$ , apabila lebih kecil  $\rho$  masukkan data (i) ke kluster yang sama, apabila tidak masukkan data (i) ke kluster baru.

Algoritma Minfirst mencari jarak data (i) dengan semua data dalam setiap kluster yang sudah ada, apabila jarak minimal lebih kecil  $\rho$  masukkan data(i) ke kluster yang sama. (Kluster yang pertama kali ditemui).

Algoritma MinBest mencari jarak data (i) dengan semua data dalam setiap kluster yang sudah ada, ambil jarak minimalnya dari setiap kluster, cari jarak minimalnya, apabila jarak minimal lebih kecil  $\rho$  masukkan data(i) ke kluster yang sama.

Algoritma MaxFirst mencari jarak data (i) dengan semua data dalam setiap kluster yang sudah ada, mencari jarak maksimalnya, apabila jarak maksimal lebih kecil  $\rho$  masukkan data(i) ke kluster yang sama (kluster yang pertama kali ditemui).

Algoritma Maxbest, mencari jarak data (i) dengan semua data dalam setiap kluster yang sudah ada, ambil jarak maksimalnya dari setiap kluster, cari jarak minimalnya, apabila jarak minimal lebih kecil  $\rho$  masukkan data (i) ke kluster yang sama.

Algoritma AveFirst masukkan data pertama ke kluster 1. Rata-rata kluster1 = data 1, mencari jarak data (i) dengan jarak rata-rata setiap kluster yang sudah ada. Apabila jarak titik(i) dengan jarak rata-rata kluster lebih kecil  $\rho$  masukkan data(i) ke kluster yang sama (kluster yang pertama kali ditemui).

Algoritma AveBest Masukkan data pertama ke kluster 1. Rata-rata kluster1 = data 1, mencari jarak data (i) dengan jarak rata-rata setiap kluster yang sudah ada, mencari jarak minimalnya. Apabila jarak lebih kecil  $\rho$  masukkan data(i) ke kluster yang sama.

## II. METODE PENELITIAN

Metode penelitian yang digunakan dalam penelitian ini adalah studi pustaka dengan dukungan implementasi program komputasi. Dengan studi pustaka diharapkan diperoleh berbagai informasi yang berhubungan dengan analisis kluster dan optimisasi. Sedangkan penggunaan program dimaksudkan untuk membuktikan kebenaran teori.

Secara rinci metode penelitian adalah sebagai berikut :

1. Memilih titik-titik yang dibangkitkan secara random.
2. Menerapkan kedelapan algoritma Heuristic untuk mengelompokkan titik-titik yang sudah didapatkan dari langkah 1.
3. Memilih satu titik terbaik sebagai wakil dari setiap klusternya.
4. Dengan menggunakan metode Newton Raphson dua variabel untuk mencari titik minimum fungsi berdasarkan titik awal yang didapat pada langkah 3.
5. Membandingkan nilai-nilai minimum yang didapat pada langkah 4 untuk mendapatkan minimum global.

Data berupa titik pasangan  $(x_1, x_2)$  dengan nilai random adalah bilangan real dari -1 sampai 1. Dilanjutkan dengan proses klustering dengan menggunakan kedelapan algoritma yang sudah ditentukan. Apabila proses klustering sudah selesai, maka dilanjutkan dengan proses pemilihan titik yang akan mewakili masing-masing kluster. Pemilihan ini dimulai dengan mencari nilai fungsi pada masing-masing titik. Nilai fungsi yang paling minimum dalam satu kluster yang dipilih untuk mewakili kluster tersebut. Apabila dalam satu kluster hanya mempunyai satu anggota atau satu titik, maka titik itulah yang menjadi wakilnya.

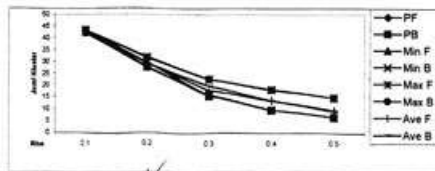
Setelah proses pemilihan wakil masing-masing kluster didapatkan, maka dilanjutkan dengan proses pengujian. Dalam proses pengujian digunakan algoritma Newton Raphson dua variabel. Dalam function ini sekaligus akan menghitung berapa banyak iterasi Newton dalam setiap algoritmanya.

### III. HASIL PENELITIAN DAN PEMBAHASAN

Percobaan dilakukan dengan menggunakan processor Intel PIII 750/100 RAM 64 MB. Data yang digunakan 50 data random yang diulang sebanyak sepuluh kali pada masing-masing rho, sedangkan rho yang digunakan adalah 0,1, 0,2, 0,3, 0,4 dan 0,5. Dari hasil penelitian akan dilihat jumlah kluster yang terbentuk, flops, waktu akses, jumlah titik minimum, banyaknya iterasi dan efektivitas.

#### III.1. Jumlah Kluster

Perbandingan jumlah kluster dengan rho pada masing-masing algoritma dapat dilihat pada gambar 1.



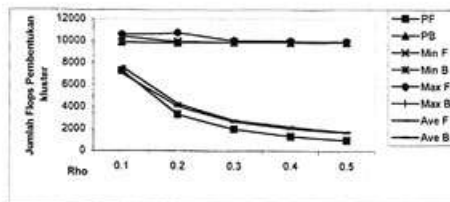
Gambar 1. Grafik perbandingan jumlah kluster

Dari gambar 1 terlihat bahwa jika rho semakin kecil jumlah kluster akan semakin banyak. Untuk rho yang sama algoritma MaxFirst dan Max Best akan membentuk jumlah kluster paling banyak, sedangkan MinFirst dan MinBest paling sedikit.

Dari gambar 1 terlihat bahwa jika rho semakin kecil jumlah kluster akan semakin banyak. Untuk rho yang sama algoritma MaxFirst dan MaxBest akan membentuk jumlah kluster paling banyak, sedangkan algoritma MinFirst dan MinBest paling sedikit.

### III.2. Flops

Perbandingan banyaknya flops dengan rho dari masing-masing algoritma dapat dilihat gambar 2.

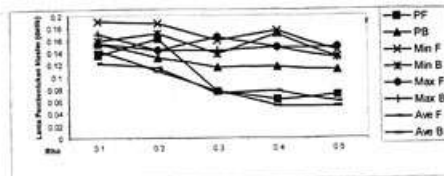


Gambar 2. Perbandingan flops pembentukan kluster

Berdasarkan gambar 2 terlihat jelas jumlah flops algoritma PointFirst paling sedikit kemudian disusul algoritma AveFirst dan AveBest sedangkan rata-rata flops paling banyak pada algoritma MaxFirst.

### III.3. Waktu Akses Pembentukan Kluster

Hasil percobaan dengan mencatat waktu pembentukan kluster berdasarkan rho pada masing-masing algoritma dapat dilihat pada gambar 3.



Gambar 3. Grafik waktu pembentukan kluster

Dalam penelitian ini kecepatan pembentukan kluster akan dilihat berdasarkan banyaknya flops dan lama pembentukan kluster. Penulis melihat pencatatan waktu akses kurang begitu akurat, ini dapat dilihat dari grafiknya yang tidak beraturan. Kemungkinan dipengaruhi karena sistem operasi yang digunakan under window ( multi tasking ), tetapi ini sudah cukup mewakili untuk melihat waktu akses masing-masing algoritma.

Rho semakin kecil waktu pembentukan kluster akan semakin lama, karena jika rho kecil jumlah kluster yang terbentuk semakin banyak. Dalam penelitian ini kecepatan pembentukan kluster akan dilihat berdasarkan banyaknya flops dan lama pembentukan kluster. Algoritma PointFirst paling cepat, hal ini disebabkan karena algoritma Pointfirst akan mengecek jarak antar titik yang pertama kali ketemu langsung masuk dalam satu kluster. Algoritma AveFirst dan AveBest cenderung cepat karena pengecekan jarak mirip algoritma PointFirst, yang membuat sedikit lebih lama karena algoritma ini harus mencari jarak rata-rata antar kluster terlebih dahulu. Algoritma MinFirst paling lama, hal ini disebabkan dalam algoritma MinFirst pengecekan

harus mencari jarak yang paling minimum dari semua kluster yang sudah terbentuk, kemudian mencari jarak minimum pertama kali ketemu.

#### 1. Titik Minimum

Titik minimum fungsi Rastrigin rho 0,1 semua algoritma menghasilkan jumlah titik minimum yang sama yaitu sekitar 39 sampai 41 titik minimum. Jika rho semakin besar algoritma MaxFirst dan MaxBest menghasilkan titik minimum paling banyak ( 15 titik minimum ), sedangkan algoritma PointFirst, PointBest, MinFirst dan MinBest mempunyai titik minimum sedikit ( 7 titik minimum).

Jika rho semakin besar jumlah titik minimum semakin sedikit, karena jika rho besar jumlah kluster sedikit. Jadi jika jumlah kluster banyak kemungkinan jumlah titik minimum akan semakin banyak.

Percobaan yang sudah dilakukan hanya ditemukan satu titik minimum, hasil ini sesuai dengan Rao (1977) bahwa fungsi Rosenbrock akan menghasilkan satu titik minimum di titik (1,1) dengan nilai  $f_{min} = 0$ . Titik minimum pada fungsi Rosenbrock tidak dipengaruhi oleh banyaknya kluster.

Perbandingan banyaknya titik minimum pada fungsi Goldstein membentuk pola yang sama dengan fungsi Rastrigin. Rho kecil semua algoritma mempunyai jumlah titik minimum yang sama. Rho besar jumlah titik minimum akan semakin sedikit. Rho kecil semua algoritma menghasilkan jumlah titik minimum sama yaitu sekitar 7 titik minimum. Semakin besar rho jumlah titik minimum semakin sedikit. Algoritma MaxFirst dan MaxBest menghasilkan titik minimum paling banyak yaitu sekitar 7 titik minimum, sedangkan algoritma PointFirst, PointBest, MinFirst dan MinBest mempunyai titik minimum sedikit ( 5 titik minimum).

#### III.5. Iterasi Newton

Banyaknya iterasi Newton pada fungsi Rastrigin dari delapan algoritma relatif konstan, yang membedakan hanya pada saat rho 0,5 algoritma PointBest mempunyai jumlah iterasi lebih sedikit dibanding dengan algoritma yang lainnya. Secara umum rata-rata iterasi = 3. Pada fungsi Rastrigin rho tidak berpengaruh terhadap banyaknya iterasi Newton.

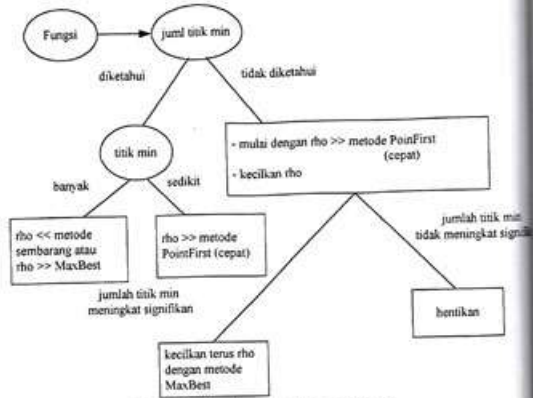
Banyak iterasi Newton untuk rho 0,1 dari semua algoritma jumlah iterasi Newton sama. Rho semakin besar jumlah iterasi akan semakin banyak. Rho 0,5 algoritma MaxBest dan MaxFirst mempunyai jumlah iterasi paling sedikit, sedangkan algoritma PointFirst mempunyai jumlah iterasi paling banyak.

Pola grafik perbandingan iterasi Newton pada fungsi Goldstein sama dengan pola pada fungsi Rastrigin. Jumlah iterasi

Newton pada fungsi Goldstein relatif konstan. Pada rho 0,5 algoritma MinBest mempunyai jumlah iterasi paling sedikit ( 6 iterasi) dan yang paling banyak algoritma MaxFirst ( sekitar 7 sampai 8 iterasi).

### III.6. Efektivitas

Secara umum bila rho semakin besar efektivitas akan semakin besar. Besarnya kenaikan tergantung dari fungsi yang digunakan. Pada fungsi Goldstein efektivitas paling tinggi sekitar 70%. Ini lebih kecil dari fungsi Rastrigin. Dilihat dari ketiga fungsi yang digunakan pada penelitian ini, dapat disimpulkan bahwa fungsi Rastrigin dan fungsi Goldstein cocok digunakan untuk metode klustering.



Gambar 5. Rangkuman hasil & pembahasan

Apabila ada sebuah fungsi yang jumlah titik minimumnya diketahui dan berjumlah banyak maka metode klustering yang paling cocok adalah



algoritma MaxBest dengan rho besar, atau sembarang algoritma dengan rho kecil. Apabila fungsi tersebut jumlah titik minimumnya sedikit maka metode PointFirst atau PointBest paling sesuai digunakan dengan rho besar, karena metode ini paling cepat.

Apabila ada sebuah fungsi yang jumlah titik minimumnya belum diketahui, maka algoritma yang pertama kali dicoba adalah algoritma PointFirst dengan rho besar, kemudian rho diperkecil. Apabila jumlah titik minimumnya meningkat signifikan kecilkan terus rho atau gunakan algoritma MaxBest. Apabila jumlah titik minimum tidak meningkat signifikan hentikan. Ini berarti fungsi tersebut tidak cocok untuk metode klustering.

#### IV. Kesimpulan

1. Rho semakin kecil, jumlah kluster akan semakin banyak.
2. Rho semakin kecil, waktu pembentukan kluster semakin lama.
3. Titik minimum fungsi Rosenbrock tidak dipengaruhi oleh banyaknya kluster, fungsi Rastrigin dan Goldstein membentuk pola yang sama. Rho kecil semua algoritma menghasilkan jumlah titik minimum sama. Semakin besar harga rho semakin sedikit jumlah titik minimumnya.
4. Banyaknya iterasi Newton untuk fungsi Rastrigin dan Goldstein relatif konstan, sedangkan untuk fungsi Rosenbrock rho semakin besar banyaknya iterasi akan semakin besar.
5. Secara umum rho semakin besar efektivitas akan semakin besar, besarnya kenaikan tergantung dari fungsi yang digunakan.

#### DAFTAR PUSTAKA

1. Anderberg, M.R., 1973. *Cluster Analysis For Applications*, Academic Press, New York.
2. Anderson, T.W., 1958. *An Introduction to Multivariate Statistical Analysis*, Wiley New York.
3. Arthanari, T.S. and Dodge Yadolah, 1981. *Mathematical Programming In Statistics*, John Wiley & Sons, Inc.
4. Becker, R.W. 1978. *A Global Optimization algorithm*, Monticello Illinois.
5. Bronson Richard, 1983. *Operation Research*, McGraw Hill, Inc.
6. Grosej Irene, 1996. *Global Optimization by Clustering an Algorithm*: Karl-Frankens-Univeritat Graz.
7. Hanselman, D., 1997. *The student Edition of MATLAB* : Printice Hall, Inc.
8. Hartigan, J.A., 1975. *Clustering Algorithms*, Wiley, New York, <http://>