

PENGUJIAN ALGORITMA *CLUSTERING* AFFINITY PROPAGATION DAN ADAPTIVE AFFINITY PROPAGATION TERHADAP IPK DAN JARAK RUMAH

Millati Izzatillah¹, Achmad Benny Mutiara²
Program Studi Informatika Universitas Indraprasta PGRI¹,
Fakultas Ilmu Komputer Universitas Gunadarma²
mizzatillah@gmail.com¹, amutiara@staff.gunadarma.ac.id²

Submitted March 10, 2020; Revised March 2, 2020; Accepted March 30, 2020

Abstrak

Clustering adalah metode untuk mengklasifikasikan data dengan mudah yang memiliki tujuan untuk melihat korelasi antara atribut data. *Clustering* juga merupakan proses pengelompokan titik data berdasarkan nilai kesamaan (*similarity*) untuk menentukan pusat *cluster*. Affinity Propagation (AP) dan Adaptive Affinity Propagation (Adaptive AP) adalah algoritma *clustering* yang menghasilkan jumlah *cluster*, anggota *cluster*, dan *exemplar* dari setiap *cluster*. Penelitian ini dibuat untuk mengetahui algoritma yang lebih efektif dalam mengelompokkan data. Selain itu untuk mengetahui perbaikan yang ditawarkan Algoritma Adaptive AP yang merupakan pengembangan dari Algoritma AP. Peneliti melakukan implementasi dan pengujian kedua algoritma dengan Matlab R2013a 8.10 menggunakan data mahasiswa berdasarkan nilai IPK (Indeks Prestasi Kumulatif) dan jarak rumah mahasiswa ke kampus sebanyak 250 data. Hasil pengujian aplikasi dianalisa cara kerja kedua algoritma dan menemukan algoritma terbaik dalam mengelompokkan data secara optimal adalah Adaptive AP. Hasil lainnya adalah tidak ada korelasi antara IPK dan jarak rumah mahasiswa ke kampus.

Kata Kunci: Algoritma *Clustering*, Pengelompokan Data, Adaptive Affinity Propagation, Affinity Propagation, Matlab.

Abstract

Clustering which is a method to classify data easily is used for a purpose of looking at the correlation among data attributes. *Clustering* is also a data point grouping based on similarity value to determine the cluster center. Affinity Propagation (AP) and Adaptive Affinity Propagation (Adaptive AP) are clustering algorithms that produce number of cluster, cluster members and exemplar of each cluster. This research is conducted to find out a more effective algorithm when clustering data. Besides, to know the correction offered by Adaptive AP Algorithm which is the developed form of AP Algorithm, the researcher implemented and tested both algorithms by using Matlab R2013a 8.10 with 250 data taken from students' GPA and the distance from their houses to campus. The analysis of test result application from both algorithms shows that the best algorithm is Adaptive AP because it produces optimal clustering. Another result is no correlation between GPA and home distance.

Keywords: Clustering Algorithm, Data Clustering, Adaptive Affinity Propagation, Affinity Propagation, Matlab

1. PENDAHULUAN

Clustering secara tradisional merupakan masalah mendasar dalam *data mining* [1]. *Clustering* adalah metode pengelompokan data berdasarkan nilai kesamaan (*similarity*) antara titik data untuk menentukan pusat *cluster* [2].

Affinity propagation (AP) adalah algoritma *clustering* yang memiliki tingkat kesalahan rendah, kecepatan tinggi dengan pengelompokan fleksibel dan sederhana [3]. Menurut B. J. Frey, Dueck, 2007 Affinity Propagation (AP) bekerja dengan "message passing" [4], yang menguji kemungkinan semua titik data menjadi

pusat cluster (*exemplar*). Hasil Affinity Propagation (AP) adalah jumlah cluster, anggota cluster dan masing-masing pusat cluster (*exemplar*).

Seiring dengan perkembangannya, Affinity propagation (AP) tidak selalu menghasilkan cluster yang ideal. Affinity propagation (AP) memiliki dua keterbatasan: (1) AP sulit untuk mengetahui nilai parameter "*preference*" sehingga sulit untuk menghasilkan solusi pengelompokan yang optimal. (2) Osilasi tidak dapat dihilangkan secara otomatis jika terjadi [5]. Keterbatasan AP akan menghasilkan cluster yang belum optimal. Untuk mengatasi keterbatasan ini, Affinity propagation (AP) dikembangkan menjadi Adaptive Affinity propagation (disebut Adaptive AP) [6].

Menurut Kaijun Wang, 2007 Adaptive AP menghilangkan osilasi yang terjadi dengan mengeliminasi (disebut *adaptive damping*). Jika *adaptive damping* gagal, Adaptive AP akan menghindari osilasi (disebut *adaptive escaping*) dan setelah itu akan dilakukan *adaptive searching* untuk menemukan solusi *clustering* yang paling optimal (disebut *adaptive scanning*) [5].

Oleh karena itu, penulis perlu menguji dan mengimplementasikan algoritma AP dan algoritma Adaptive AP serta membandingkan kedua algoritma tersebut untuk mengetahui seberapa efektif algoritma tersebut dalam mengelompokkan data. Tes yang dilakukan menggunakan data mahasiswa Universitas Gunadarma berdasarkan Indeks Prestasi Kumulatif (IPK) dan jarak dari rumah mahasiswa menuju kampus yang diimplementasikan menggunakan *software* Matlab versi R2013a 8.10.

2. METODE PENELITIAN

Pengumpulan Data

Pengumpulan data adalah tahap untuk mengumpulkan data yang akan digunakan

untuk menguji algoritma dalam penelitian. Data yang digunakan dalam penelitian ini adalah data yang diperoleh dari hasil *survey* kepada mahasiswa Universitas Gunadarma dengan cara melihat Kartu Hasil Studi untuk mendapatkan nilai IPK dan Kartu Tanda Mahasiswa untuk mendapatkan alamat tempat tinggal.

Jumlah data yang digunakan 250 mahasiswa (minimum data untuk *clustering*) [7]. Data diambil secara acak tanpa melihat latar belakang jurusan responden. Dari 250 responden yang telah mengisi kuesioner diperoleh 38% jurusan sistem informasi, 23% jurusan teknik informatika, 10% jurusan akuntansi, 10% jurusan psikologi, 6% jurusan manajemen, 4% jurusan sistem komputer, 2% jurusan arsitektur, 2% jurusan teknik listrik, 2% jurusan manajemen informatika, 2% jurusan sastra inggris, 1% jurusan teknik mesin. Data yang digunakan dalam penelitian ini hanya dua variabel, Indeks Prestasi Kumulatif (IPK) dan jarak rumah mahasiswa. Jarak rumah ditempuh dengan jarak dari alamat rumah ke kampus Gunadarma di Margonda, Depok.

Analisa Kebutuhan

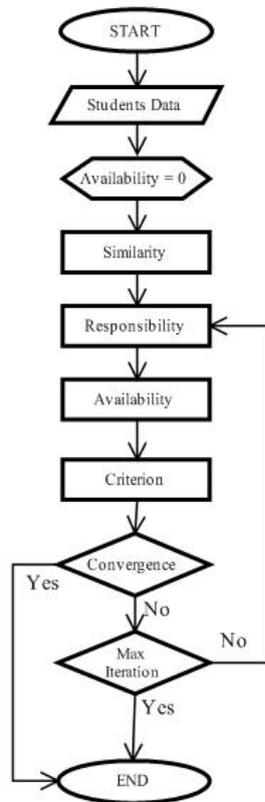
Analisa kebutuhan adalah tahap untuk menjelaskan kebutuhan fungsional aplikasi kebutuhan tersebut adalah:

1. Kemampuan untuk mengelompokkan data dengan dua algoritma, AP dan Adaptive AP.
2. Kemampuan untuk menghasilkan jumlah *cluster* dan menunjukkan *exemplar*-nya.
3. Kemampuan untuk *plotting* hasil *cluster*.
4. Kemampuan untuk menghitung *runtime* algoritma AP dan Adaptive AP bekerja.

Affinity Propagation

Affinity propagation bekerja melalui beberapa tahap. Tahapan-tahapan ini

dijelaskan oleh diagram alur program pada Gambar 1.



Sumber : Dokumentasi Penulis

Gambar 1. Program Flowchart Affinity Propagation

Data mahasiswa yang digunakan dalam penelitian ini adalah Indeks Prestasi Kumulatif (IPK) dan jarak rumah sebanyak 250 mahasiswa. Data yang masuk adalah 250 data, kemudian data tersebut akan menjadi matriks s (250×250).

Similarity menunjukkan nilai kesesuaian titik data ke titik data lainnya. *Similarity* dilambangkan dengan $s(i,k)$, i menunjukkan baris dan k sebagai kolom, yang menunjukkan seberapa cocok titik data i menunjuk ke titik data k .

Nilai *similarity* yang digunakan dalam algoritma AP adalah nilai negatif jarak setiap titik data. Perhitungan jarak menggunakan teknik *Euclidean Distance*. Penulis mengambil lima *record* dari data untuk memberikan contoh menemukan

nilai *similarity*. Data dapat dilihat pada tabel 1.

Tabel 1. Lima record dari data mahasiswa

No	IPK (x)	Jarak Rumah (KM) (y)
1	3.75	15
2	3.68	25
3	3.55	15
4	3.71	0.1
5	2.62	5

Sumber : Dokumentasi Penulis

Perhitungan *similarity* dapat dilihat sebagai berikut:

$$S(1,2) = \sqrt{|x_1(1) - x_1(2)|^2 - (y_2(1) - y_2(2))^2} \quad (1)$$

$$S(1,2) = \sqrt{|3.75 - 3.68|^2 - (15 - 25)^2}$$

$$S(1,2) = -10.0002$$

Setelah setiap $s(i, k)$ dihitung didapatkan matriks sebagai berikut.

Tabel 2. Similarity Matrix

	1	2	3	4	5
1	0	-10.0002	-0.2000	-14.9001	-10.0636
2	-10.0002	0	-10.0008	-24.9000	-20.0281
3	-0.2000	-10.0008	0	-14.9000	-10.0432
4	-14.9001	-24.9000	-14.9009	0	-5.0198
5	-10.0636	-20.0281	-10.0432	-5.0198	0

Sumber : Dokumentasi Penulis

Nilai diagonal tabel di atas adalah 0, nilai harus diganti dengan nilai yang disebut "*preference*". Nilai *preference* yang digunakan adalah nilai minimum atau nilai median *similarity* yang terbentuk. Minimum *preference* menggunakan nilai minimum yaitu -24,9000 dan Median *preference* menggunakan nilai tengah yaitu -10,0534. Jadi isi diagonal dengan nilai preferensi minimum.

Setelah mendapatkan *similarity matrix*, langkah selanjutnya adalah menghitung nilai *responsibility*. Perhitungan *responsibility matrix* adalah sebagai berikut.

$$R(1,2) \leftarrow S(1,2) - \max\{a(1,2') + S(1,2')\} \quad (2)$$

Setelah setiap $r(i,k)$ dihitung dengan *minimum preference* adalah sebagai berikut:

Tabel 3. Responsibility Matrix Minimum Preference

	1	2	3	4	5
1	-12.3500	-4.9001	4.9001	-7.3500	-4.9318
2	0.0003	-7.4499	-0.0003	-7.4499	-5.0139
3	4.9004	-4.9004	-12.3500	-7.3504	-4.9216
4	-4.9401	-9.9401	-4.9405	-9.9401	4.9401
5	-2.5219	-7.5041	-2.5117	2.5117	-9.9401

Sumber : Dokumentasi Penulis

Untuk iterasi pertama, nilai awal *availability* yang digunakan untuk menentukan nilai *responsibility* adalah nol. Setiap $a(i,k)$ and $a(k,k)$ dengan *minimum preference* dihitung dan didapatkan matriks sebagai berikut.

Tabel 4. Availability Matrix Minimum Preference

	1	2	3	4	5
1	2.4504	-3.7249	-6.1750	-3.7142	-2.5000
2	-3.7248	0	-3.7249	-3.7142	-2.5000
3	-6.1749	-3.7249	2.4501	-3.7142	-2.5000
4	-3.7246	-3.7249	-3.7249	1.2558	-4.9701
5	-3.7246	-3.7249	-3.7249	-4.9701	2.4701

Sumber : Dokumentasi Penulis

Criterion adalah penjumlahan matriks dari nilai *responsibility* and *availability*. Diagonal *criterion* memiliki nilai positif akan menjadi *exemplar*. Titik data yang memiliki nilai *similarity* terdekat dengan *exemplar* akan berada dalam satu cluster. Jika iterasi dilakukan seratus kali untuk lima *record* dari data mahasiswa tersebut, hasil matriks *criterion* adalah sebagai berikut:

Tabel 4. Criterion Matrix Minimum Preference

	1	2	3	4	5
1	-24.6986	-24.6994	14.8394	-19.6758	-14.8394
2	-24.6986	-14.8992	14.8992	-19.8750	-15.0030
3	-39.6887	-39.6901	4.9758	-34.6668	-29.8091
4	-24.6986	-24.8983	4.9758	-4.9758	-4.9758
5	-24.7191	-24.8841	4.9758	-4.9758	-4.9758

Sumber : Dokumentasi Penulis

Kolom ketiga pada matriks *criterion minimum preference* memiliki nilai positif yaitu 4,9758, maka akan ada satu *cluster* dengan data ketiga menjadi *exemplar*. Semua titik data memiliki nilai *similarity* yang hampir sama dengan titik data 3. Jadi, titik data 1, 2, 4 dan 5 berada dalam satu cluster di mana *exemplar*-nya adalah titik data 3.

Setelah mendapatkan jumlah *cluster* dari setiap iterasi, tahap selanjutnya adalah menentukan apakah jumlah *clustering* optimal. Jumlah *clustering* optimal akan terjadi ketika mencapai nilai konvergen. Nilai konvergen akan dicapai dengan beberapa kondisi sebagai berikut:

- Jumlah *cluster* yang dihasilkan akan selalu sama setelah iterasi minimum terjadi. Iterasi minimum adalah 50 kali iterasi [4].
- *Exemplar* (pusat *cluster*) tidak berubah dalam beberapa kali iterasi.

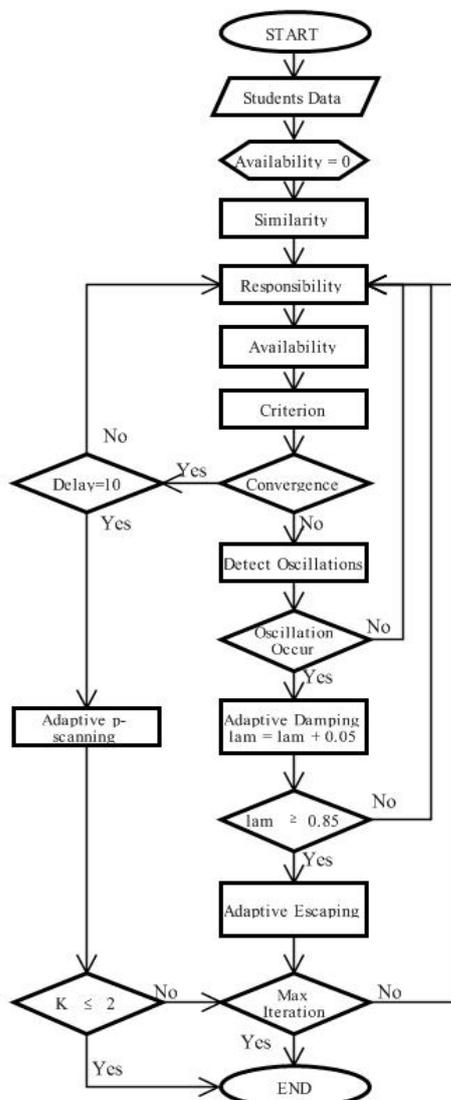
Jika iterasi mencapai konvergen pada iterasi tertentu, maka algoritma berhenti, jika tidak konvergen, maka akan memeriksa apakah iterasi telah mencapai iterasi maksimum ($maxit = 50000$, $maxit$ adalah iterasi maksimal [5]).

Jika tidak terjadi konvergen, maka kembali ke *responsibility*. Jika algoritma telah mencapai iterasi maksimum akan berhenti dan menghasilkan jumlah cluster meskipun tidak mencapai nilai konvergen.

Adaptive Affinity Propagation

Adaptive AP adalah pengembangan algoritma AP yang mengusulkan peningkatan untuk mencapai hasil pengelompokan yang lebih optimal dan mengatasi keterbatasan AP. Oleh karena itu, beberapa tahap awal dari AP Adaptif dan AP sama dengan AP yaitu (1) input, (2) *similarity*, (3) *responsibility*, (4) *availability* dan (5) *criterion*.

Adaptive Affinity propagation (Adaptive AP) bekerja melalui beberapa tahap yang dijelaskan oleh diagram alur program pada Gambar 2.



Sumber : Dokumentasi Penulis

Gambar 2. Program Flowchart Adaptive Affinity Propagation

Setelah mendapatkan jumlah *cluster* dari setiap iterasi, tahap selanjutnya adalah menentukan apakah jumlah *cluster* mencapai optimal. Jumlah *cluster* optimal akan terjadi ketika mencapai nilai konvergen. Nilai konvergensi akan dicapai dengan beberapa kondisi sebagai berikut:

- Jumlah *cluster* akan selalu sama setelah iterasi ke- v kali (50 iterasi), (*window size* = 40, *plus delay* = 10).
- *Exemplar* (pusat *cluster*) tidak berubah setelah sekian kali iterasi.

In Adaptive AP, setelah memeriksa konvergen, ada beberapa tahapan sebelum memeriksa iterasi maksimum (*maxit* = 50000 iterasi maksimal [5]). Jika k mencapai konvergen dengan iterasi tambahan dy (*delay* = 10) akan melakukan *adaptive p-scanning*.

Jika tidak konvergen dengan iterasi tambahan (*delay* \neq 10) kemudian akan melanjutkan iterasi *responsibility*. Jika tidak juga mencapai konvergen makan akan mengecek keberadaan isolasi.

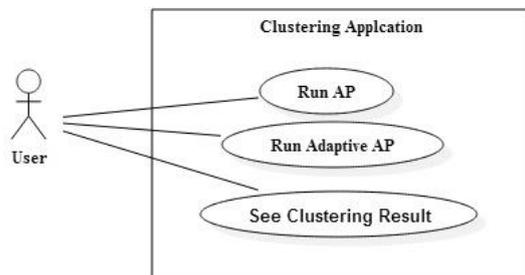
Mendeteksi osilasi terlalu rumit untuk dijelaskan. Jadi deteksi osilasi dengan mendefinisikan fitur non-osilasi. Kondisi non-osilasi ketika *exemplar* menurun atau tidak berubah selama proses iterasi. *Exemplar* yang mengalami penurunan atau perubahan akan terjadi konvergensi. Osilasi terjadi ketika jumlah *exemplar* yang dihasilkan naik dan berubah 2/3 dari *window size* maka dari itu lambda pun akan meningkat. Nilai awal lambda adalah 0.5. Teknik ini akan bekerja ketika lambda telah mencapai 0,85 atau lebih tinggi.

Lambda yang besar membawa efek yang kecil menunjukkan bahwa osilasi adalah terjadi di bawah p yang diberikan, jadi alternatifnya adalah mengurangi p dari p yang diberikan untuk menjauhkan dari osilasi. *Adaptive Escaping* dirancang untuk menghindari osilasi ketika Adaptive Damping gagal untuk menekan osilasi.

Jika mencapai iterasi maksimum maka algoritma berhenti, jika tidak pergi ke *responsibility*.

Disain Aplikasi

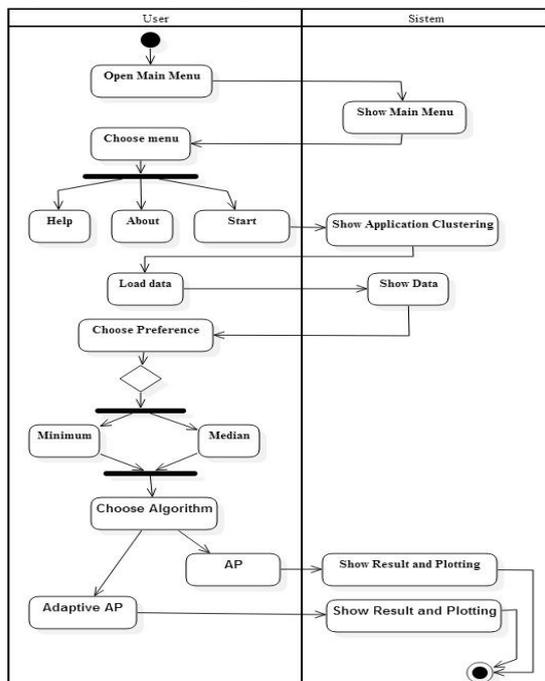
Desain Aplikasi menggunakan UML (*Unified Modeling Language*) yang terdiri dari diagram *Use Case* dan diagram *Activity* [8]. Diagram *Use Case* menggambarkan kasus, aktor dan hubungan antara kasus dan aktor. Use case untuk aplikasi pengelompokan AP dan Adaptive AP dapat dilihat pada gambar 3.



Sumber : Dokumentasi Penulis

Gambar 3. Diagram Use Case

Activity diagram menggambarkan aktivitas yang terjadi ketika pengguna menjalankan aplikasi. Tahapan proses diagram aktivitas dapat dilihat pada gambar 4.



Sumber : Dokumentasi Penulis

Gambar 4. Activity Diagram

Pada gambar 4 *activity diagram*, pengguna membuka menu utama dan aplikasi akan menampilkan menu utama. Setelah menu utama muncul, ada tiga kegiatan yang bisa dilakukan oleh pengguna, ada buka menu bantuan, buka menu dan buka menu mulai untuk menjalankan aplikasi. Ketika pengguna memilih untuk menjalankan aplikasi, aplikasi akan menampilkan *clustering* menu. Setelah itu, pengguna memuat data dan aplikasi akan menampilkan data yang akan digunakan dalam *clustering*. Pengguna memilih *preference* dan algoritma yang digunakan sebelum *clustering*. Jika pengguna memilih untuk menjalankan algoritma AP maka aplikasi akan menampilkan hasil *clustering* dan *plotting* ke algoritma AP. Namun, jika pengguna memilih untuk menjalankan algoritme Adaptive AP maka aplikasi akan menampilkan hasil *clustering* dan *plotting* untuk algoritma Alaptive AP.

3. HASIL DAN PEMBAHASAN

Implementasi

Aplikasi *clustering* ini adalah berbasis desktop yang diimplementasikan dalam Matlab 8.10 R2013a. Implementasi aplikasi ini dimulai dari pembuatan halaman menu utama, pengelompokan halaman, halaman bantuan, tentang halaman, halaman output AP dan halaman output Adaptive AP. Membuat antarmuka di Matlab menggunakan Matlab GUIDE dan menyimpan file dengan ekstensi *.fig* dan file *.m* akan mengikuti secara otomatis. Kode atau fungsi tambahan akan ditulis dalam file *.m*.

Berikut ini adalah tampilan aplikasi yang didesain:

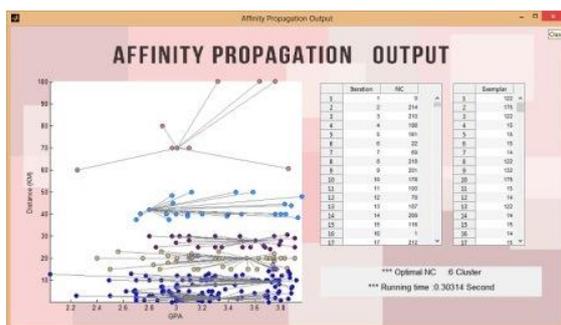


Source : Author Documentation

Gambar 5. Menu Utama Aplikasi Clustering

Saat menekan tombol *load data* seperti pada gambar 5, aplikasi akan menampilkan data ke dalam tabel untuk *clustering*. Setelah data dimuat, data akan dikelompokkan dengan mengklik tombol *cluster*. Tombol *cluster* bekerja setelah memilih *preference* (minimum atau median) dan algoritma (AP atau Adaptive AP) yang akan digunakan dalam *clustering*.

Ketika tombol *clustering* di klik pada gambar 5, aplikasi akan menampilkan hasil *clustering*, ada plot, teks dan tabel seperti gambar 6 di bawah ini.

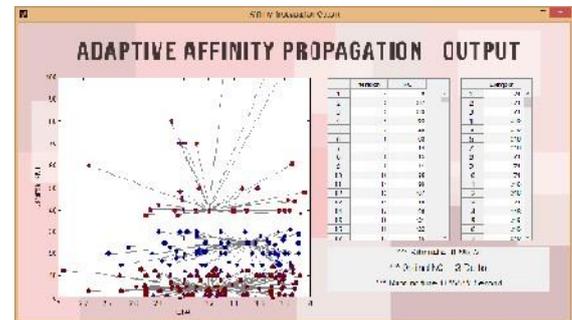


Sumber : Dokumentasi Penulis

Gambar 6. Hasil Clustering AP

Pada Gambar 6 hasil *clustering* AP, terdapat sumbu untuk menampilkan hasil plot, tabel iterasi dan tabel jumlah *cluster*, tabel *exemplar* setiap iterasi, dan teks Optimal NC (*clustering optimal*) dan Runtime dalam *clustering* data.

Adaptive Affinity Propagation dapat dilihat pada gambar 7 di bawah ini.



Sumber : Dokumentasi Penulis

Gambar 7. Hasil Clustering AAP

Pada gambar 7 adalah hasil *Clustering* AAP sama seperti AP, terdapat sumbu untuk menampilkan hasil plot, tabel iterasi dan tabel jumlah *cluster*, tabel *exemplar* setiap iterasi, dan teks Optimal NC (*clustering optimal*) dan Runtime dalam *clustering* data.

Pengujian Algoritma

Aplikasi telah dibuat dalam file yang dapat dieksekusi dan siap digunakan pada komputer tanpa proses instalasi Matlab. Jumlah data yang diuji 250 *record*. Langkah-langkah pengujian dijelaskan sebagai berikut:

- Aplikasi dijalankan dengan mengklik dua kali pada file Aplikasi Clustering. exe. Kemudian, aplikasi akan menampilkan halaman menu utama.
- Klik tombol mulai untuk masuk ke menu *clustering*. Maka aplikasi akan menampilkan halaman *clustering*. Klik tombol load untuk menampilkan data.
- Dalam proses *clustering*, diperlukan dengan memilih nilai *preferences* dan algoritma pada *combo box* di aplikasi.
- Pengujian pertama menggunakan *preferences* minimum yang dikelompokkan oleh algoritma AP.

Kemudian tekan tombol cluster. Pengujian dilakukan tiga kali.

- Pengujian kedua menggunakan *preferences* median yang dikelompokkan oleh algoritma AP. Kemudian tekan tombol cluster. Pengujian dilakukan tiga kali.
- Pengujian ketiga menggunakan *preferences* minimum yang dikelompokkan oleh algoritma Adaptive AP. Kemudian tekan tombol cluster. Pengujian dilakukan tiga kali.
- Pengujian keempat menggunakan *preferences* median yang dikelompokkan oleh algoritma Adaptive AP. Kemudian tekan tombol cluster. Pengujian dilakukan tiga kali.

Algoritma ini telah diuji sebanyak masing-masing tiga kali [9] nilai *preferences* median dan *preferences* minimum untuk algoritma affinity propagation (AP) dan adaptive affinity propagation (Adaptive AP).

4. SIMPULAN

Tidak ada konvergensi di AP. Jadi, AP menghasilkan hasil clustering yang tidak optimal. Adaptive AP berhasil mengatasi keterbatasan AP yang menghasilkan hasil *clustering* optimal dengan mencapai konvergensi tanpa pengujian yang mencapai *maxit*. Dari hasil yang disebutkan di atas perbaikan yang diberikan kepada AP dari Adaptive AP sebagai berikut:

- Adaptive AP menghasilkan pengelompokan optimal.
- Adaptive AP menghasilkan hasil pengelompokan statis.
- Adaptive AP menunjukkan konvergensi bukan tahap akhir untuk menghentikan iterasi untuk menghasilkan *cluster* yang optimal karena ada kemungkinan osilasi terjadi dengan kedua titik data yang bergantian antara menjadi *exemplar*

atau *non-exemplar*. Adaptive AP secara otomatis menghilangkan osilasi.

- Meskipun Adaptive AP memiliki lebih banyak baris kode tetapi dalam salah satu pengujian Adaptive AP Runtime lebih cepat daripada dengan AP. Itu karena AP melakukan iterasi hingga maksimal. Semakin banyak iterasi, runtime akan lebih lama.
- Jumlah optimal *cluster* dihasilkan oleh Adaptive AP dari NC yang memiliki nilai siluet tertinggi. Nilai siluet digunakan untuk mengevaluasi NC yang sesuai untuk menghasilkan hasil clustering yang optimal.

Hasil algoritma *clustering* affinity propagation (AP) dan adaptive affinity propagation (Adaptive AP) menggunakan data mahasiswa berdasarkan IPK dan jarak rumah tidak ada korelasi antara jarak rumah dengan IPK yang mereka dapatkan. Dapat disimpulkan bahwa IPK mahasiswa tidak dipengaruhi oleh jarak tempat tinggal mereka ke kampus. Dengan kata lain jarak rumah ke kampus tidak mempengaruhi prestasi belajar mahasiswa Universitas Gunadarma.

Dalam pengembangan lebih lanjut, pengujian sebaiknya dilaksanakan untuk jumlah data yang lebih besar (1000 *record*) [10] dan variabel yang digunakan untuk melihat korelasi antar data dapat meningkat lebih dari dua variabel.

DAFTAR PUSTAKA

- [1] Jian Pei Jiawei Han, Micheline Kamber. *Data Mining Concepts and Techniques*. Elsevier Inc, 2012.
- [2] Xu-qing ZHANG Yue-ting ZHUANG Ding yin XIA, Fei WU. Local and global approaches of affinity propagation clustering for large scale data. *Journal of Zhejiang University SCIENCE A*, 2008.

- [3] Precha Thavikulwat. Affinity propagation: A clustering algorithm for computer-assisted business simulations and experiential exercises. *Developments in Business Simulation and Experiential Learning*, 2008.
- [4] Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *sciencemag*, 2007.
- [5] Dan Li-Xinna Zhang Kaijun Wang, Junying Zhang and Tao Guo. Adaptive affinity propagation clustering. *Automatica Sinica*, 2007.
- [6] Jure Leskovec Anand Rajaraman Jeffrey D. Ullman. Mining of massive datasets. This book evolved from material developed over several years by Anand Rajaraman and Jeff Ullman for a one-quarter course at Stanford., Palo Alto, CA, March 2014.
- [7] Chandan K. Reddy Charu C. Aggarwal. *Data Clustering Algorithms and Applications*. CRC Press, 2014.
- [8] Tim Weilkiens and Bernd Oestereich. UML 2 Certification Guide: *Fundamental & Intermediate Exams*. Morgan Kaufmann, 2010.
- [9] Yueting Zhuang Xuqing Zhang, Fei Wu. *Clustering by evidence accumulation on affinity propagation*. IEEE, 2008.
- [10] Ali Ridho Barakbah. Clustering. Institut Teknologi Surabaya, 2006.