Implementasi Algoritma *Self Organizing Map* (SOM) untuk *Clustering* Mahasiswa pada Matakuliah Proyek (Studi Kasus : JTK POLBAN)

Ghifari Munawar

Jurusan Teknik Komputer dan Informatika, Politeknik Negeri Bandung, Bandung 40012
Email : ghifari.munawar@polban.ac.id

ABSTRAK

Matakuliah proyek merupakan matakuliah berbasis *Problem Based Learning* (PBL) dimana proses pembelajarannya dilakukan secara berkelompok. Selama ini proses pengelompokkan masih dilakukan secara manual, dan setiap kelompok diharapkanmemiliki anggota dengan kemampuan akademis yang sama. Penelitian ini bertujuan mengimplementasikan algoritma *Self Organizing Map* (SOM) untuk *clustering* mahasiswa pada matakuliah proyek, dan mengevaluasi hasil *clustering* sistemnya. Parameter yang dijadikan dasar *clustering* adalah nilai-nilai matakuliah di semester berjalan. Proses *training* pada algoritma SOM dilakukan dengan mencari jarak terdekat dari masing-masing *neuron*output ke data input, proses ini akan mengupdate bobot *neuron* pada setiap iterasi hingga mencapai nilai*error* terkecil. Pengujian dilakukan dengan menguji fungsionalitas sistem, dan mengevaluasi *cluster* yang dihasilkannya. Pengujian fungsionalitas sistem dilakukan secara *blackbox* dengan beberapa butir uji, sedangkanpengujian *clustering* dilakukan dengan menghitung nilai *mean square error*(MSE)pada setiap *cluster*. Pengujian *clustering*ini dilakukansebanyak 5 kali dengan skenario dan data uji yang sama. Semakin kecil nilai MSE menunjukan *cluster* yang terbentuk semakin konvergen. Nilai rata-rata MSE yang terkecil adalah 3,2802 pada pengujian ke- 2, sedangkan yang terbesar adalah 3,5406 pada pengujian ke- 4 denganselisih nilai sebesar 0,2604. Hasil pengujian ini menunjukan bahwa*cluster* yang terbentuk memiliki anggota dengan bobot nilai yang konvergen.

Kata Kunci
Sistem Clustering, Self Organizing Map (SOM),Mean Square Error (MSE), Matakuliah Proyek

1. PENDAHULUAN

Matakuliah proyek pada Jurusan Teknik Komputer dan Informatika (JTK) POLBAN merupakan matakuliah berbasis PBL (*Problem Based Learning*), dimana metode pembelajarannya didasarkan pada permasalahan sebagai stimulus dalam memperoleh pengetahuan baru dan mengintegrasikannya dengan pengetahuan yang telah dimiliki sebelumnya. Dalam penerapannya, pembelajaran di matakuliah ini dilakukan secara berkelompok, dimana masing-masing kelompok memiliki anggota mahasiswa dengan tingkat kemampuan akademis yang cenderung sama. Tujuannya adalah untuk menumbuhkan semangat belajar bersama tanpa adanya kesenjangan kemampuan akademis, dan menghindari adanya dominasi dari salah satu/lebih anggota kelompoknya.

Selama ini proses pengelompokan pada matakuliah proyekmasih dilakukan secara manual dengan cara merangking mahasiswa berdasarkan nilai Indeks Prestasi Semester (IPS) dari yang paling besar hingga paling kecil, dari daftar tersebut kemudian ditentukan jumlah anggota perkelompoknyadan dibagi sesuai dengan urutan rangking. Pengelompokkan seperti ini belum sepenuhnya efektif dalam mendapatkan kelompok mahasiswa dengan anggota yang konvergen.

Clustering adalah proses untuk menempatkan sekumpulan *record* data kedalam satu himpunan atau kelompok yang disebut *cluster*, sehingga dalam satu *cluster* memiliki *record* data dengan karakteristik yang sama dan berbeda dengan *cluster* lainnya [1]. Permasalahan dasar dari *clustering* adalah bagaimana membagi sekumpulan data yang memiliki kesamaan semirip mungkinke dalam satu *cluster*. Ada beberapa algoritma *clustering* yang dapat digunakan, salah satunya adalah *Self Organizing Map* (SOM). SOM pertama kali diperkenalkan pada tahun 1981 oleh Prof. Teuvo Kohonen, algoritma ini melakukan proses *clustering* dengan membentuk jaringan kohonen/SOM yang digunakan untuk mengelompokkan data berdasarkan karakteristik/fitur-fitur datanya [2]. Berdasarkan penelitian yang telah dilakukan, algoritma SOM efektif digunakan untuk *clustering* dimana target outputnya tidak memerlukan pengawasan (*unsupervised*) [3].

Dasar inilah yang menjadi latar belakang dalam penelitian ini, bagaimana mengimplementasikan algoritma *Self Organizing Map* untuk mengelompokkan mahasiswa di matakuliah proyek agar mendapatkan kelompok mahasiswayang konvergen.

Pada implementasinya, sistem*clustering* dengan algoritma SOM ini akan dikembangkan menggunakan bahasa C#. Untuk mengevaluasi hasil *cluster*, tingkat kesalahan dalam proses *clustering*akan diukur menggunakan rumus *mean square error* (MSE).Semakin kecil nilai MSE menunjukan bahwa *cluster* yang terbentuk semakin konvergen.

1.1 Rumusan Masalah

Permasalahan yang diangkat dalam penelitian ini adalah *clustering* data menggunakan algoritma SOM. Data yang akan di-*cluster* adalah data-data nilai mahasiswa dari beberapa matakuliah lain yang relevan dengan matakuliah proyek di semester berjalan. Tujuan dari proses *clustering* ini adalah untuk mendapatkan *cluster* dengan anggota mahasiswa yang memiliki kemampuan akademis yang sama. Dari permasalahan tersebut, maka rumusan masalah dalam penelitian ini adalah :

1. Bagaimana mengimplementasikan*clustering* mahasiswa pada matakuliah proyek dengan algoritma SOM.
2. Bagaimana analisa hasil *clustering*nya.

1.2 Tujuan Penelitian

Berdasarkan rumusan masalah, maka tujuan dari penelitian ini adalah :

1. Mengembangkan sistem *clustering* mahasiswa pada mata kuliah proyekdenganalgoritma SOM.
2. Menganalisa*cluster* yang dihasilkan olehalgoritma SOM pada studi kasus ini.

2. TINJAUAN PUSTAKA

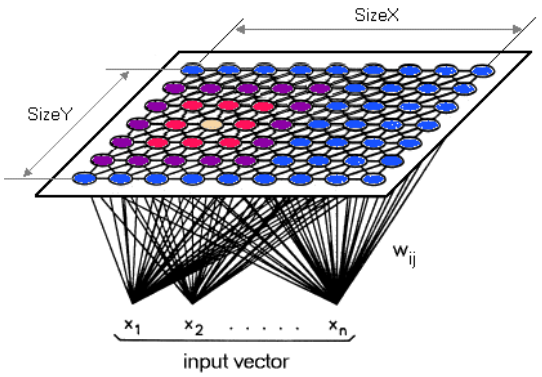
2.1 *Clustering*

Clustering adalah proses untuk menempatkan sekumpulan *record* data kedalam satu himpunan atau kelompok yang disebut *cluster*, sehingga dalam satu *cluster* memiliki *record* data dengan karakteristik yang sama dan berbeda dengan *cluster* lainnya [1].Tahapan proses *clustering* dapat dibagi menjadi 5 tahap berikut[1]:

(a) Representasi pola;
(b) Mengukur perbedaan yang terdefinisi;
(c) *Clustering*;
(d) Abstraksi data;
(e) Penilaian *Output*;

2.2 *Self Organizing Map (SOM)*

Algoritma *Self Organizing Map* (SOM) pertama kali diperkenalkan oleh Prof. Teuvo Kohonen pada tahun 1982. Kohonen *Self Organizing Map* (SOM) merupakan salah satu algoritma *clustering* yang paling populer dan merupakan salah satu *tool* visualisasi yang handal untuk memproyeksikan hubungan kompleks dari ruang input berdimensi tinggi kedalam sebuah ruang berdimensi rendah (biasanya berupa grid 2 dimensi) [3].



Gambar 1: Proses pemetaan vektor input ke dalam grid 2 dimensi [3]

Dalam kaitannya dengan penelitian ini, metode SOM akan digunakan untuk mengelompokkan mahasiswa berdasarkan data nilai dari beberapa matakuliah di semester berjalan. Data nilaimerupakan vektor input dari proses *clustering* ini, kemudian SOM akan membentuk *neuron* outputsesuai jumlah *cluster* yang diharapkan ke dalam grid 2D. Setelah proses *training* selesai,masing-masing vektor input akan dipetakan pada *cluster*sesuai dengan bobot yang terdekat.

2.3 Algoritma SOM

Berikut ini adalah tahapan algoritma *Self Organizing Map* [4, 5]:

1. Inisialisasi vektorinput x_1, x_2, x_3, ... x_n.
2. Inisialisasi *neuron* output sebanyak y_1, y_2, y_3, ... y_n.
3. Menentukan *weight* (bobot) *neuron* output dengan nilai antara x_{min} dan x_{max}.
4. Mengulangi langkah 5 sampai 8 hingga tidak ada update *weight* (bobot) atau telah mencapai kondisi stop (*error* terkecil).
5. Pemilihan acak salah satu data dari vektor input sebagai data *training*.
6. Mencari jarak terdekat dari masing-masing *neuron* output ke data inputmenggunakan rumus *euclidian distance*.

$$D_i = \sum_{i=1}^{n} (w_{ij} - x_i)^2 \qquad (1)$$

Dari seluruh bobot (D_i) dicari yang paling kecil jaraknya, indeks dari bobot (D_i) ini disebut *winning neuron*.

7. Untuk setiap bobot w_{ij} diperbaharui bobot tetangga menggunakan rumus dengan persamaan sebagai berikut :

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(t)[x_i - w_{ij}(t)] \qquad (2)$$

8. Mengupdate bobot bias (*error*).
9. Simpan bobot yang telah konvergen.

2.4 *Mean Square Error*

Mean Square Error (MSE) merupakan salah satu metode untuk mengevaluasi hasil *cluster*, metode ini mengukur tingkat kesalahan (*error*) dengan menghitung jumlah kuadratdari jarak vektor input terhadap *winning neuron* dibagi dengan jumlah bobotnya. Berikut ini adalah rumus yang digunakan untuk menghitung MSE [6]:

$$MSE = \sum_{t=1} \left(\frac{y - y^t}{n}\right)^2 \qquad (3)$$

Perhitungan MSE berfungsi untuk mengukur kesalahan inisialisasi bobot yang dilakukan secara acak saat proses *training* berlangsung, dimana keacakan data ini akan mempengaruhi tingkat konvergen. Semakin kecil nilai MSE, menunjukan bahwa tingkat konvergennya semakin baik.

2.5 *Unified Modelling Language (UML)*

UML (*Unified Modeling Language*) merupakan sebuah bahasa berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan mendokumentasikan sebuah pengembangan perangkat lunak berbasis OO (*Object-Oriented*). UML memberikan standar penulisan untuk*blue print* sistem, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam memodelkan perangkat lunak [7].Bahasa pemodelan UML lebih cocok digunakan untuk pembuatan perangkat lunak yang diimplementasikan dalam bahasa pemrograman berorientasi objek (C#, Java), namun demikian tetap dapat digunakan pada bahasa pemrograman prosedural [8].

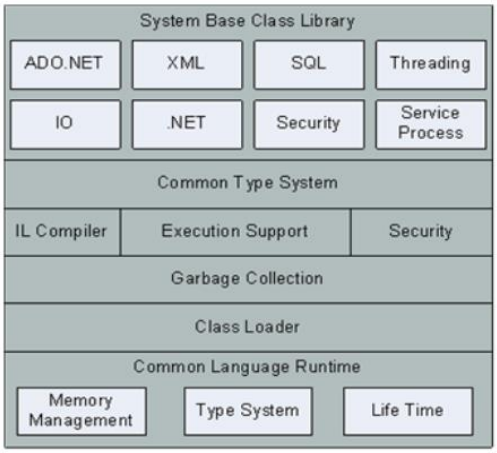
Beberapa diagram UML yang dapat digunakan untuk pemodelan, diantaranya [8] :

- *Use Case Diagram*
- *Class Diagram*
- *Sequence Diagram*
- *State Diagram*

- *Activity Diagram*
- *Deployment Diagram*

2.6 Bahasa C#

C# merupakan bahasa pemrograman berorientasi objek yang merupakan bagian bahasa pemrograman dari lingkup .NET dan memiliki akses penuh terhadap *Framework Class Library* (FCL). *Framework* .Netmerupakan sekumpulan *library* yang dapat akses untuk mempercepat pengembangan aplikasi [9]. *Framework class library* merupakan *standard library* yang tersedia untuk semua bahasa pemrograman dalam lingkup .Net yang merangkum sejumlah besar fungsi-fungsi umum, seperti membaca dan menulis (IO), rendering grafis, interaksi *database*, dan manipulasi dokumen XML [10]. Berikut ini adalah arsitektur dari *framework class library*:

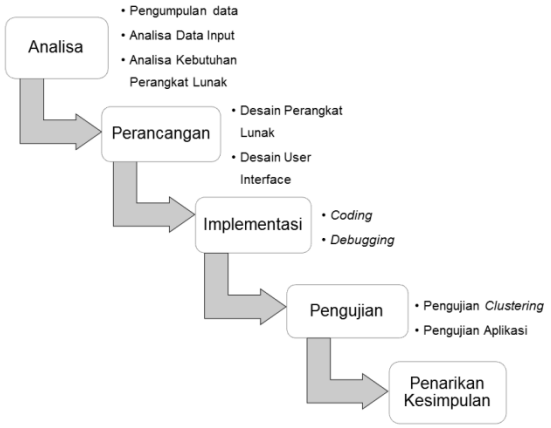


Gambar2 :Framework class library .Net[10]

3. METODE PENELITIAN

Penelitian ini dilaksanakan pada Jurusan Teknik Komputer dan Informatika Politeknik Negeri Bandung. Data-data yang digunakan adalah data nilai buku besar mahasiswa yang telah mengikuti matakuliah Proyek untuk tahun ajaran 2014/2015, serta data kurikulum yang berisi daftar matakuliah yang diberikan kepada mahasiswa di tiap-tiap semesternya.

Penelitian yang dilakukan adalah penelitian kuantitatif, artinya hasil analisis disajikan dalam bentuk angka-angka / statistik yang kemudian dijelaskan dalam bentuk uraian. Berikut ini adalah tahapan penelitian yang dilakukan dan dimodelkan dalam bentuk *waterfall* dimana masing-masing tahapan harus dilalui secara runut sebelum masuk ke tahap selanjutnya :



Gambar 3 : Tahapan Penelitian

Secara garis besar penelitian ini dibagi menjadi tiga tahapan, diantaranya(1) tahap analisa;(2) tahap perancangan dan implementasi;serta(3) tahap pengujian. Tahap analisa dilakukan untuk menganalisa data yang telah dikumpulkan dengan bekal studi terkait permasalahan yang dihadapi, tahap perancangan dan implementasi bertujuan untuk memodelkan perancangan perangkat lunak ke dalam bentuk diagram, dan membuat kode pemrograman untuk mengimplementasikan perangkat lunak yang akan dibuat. Sedangkan tahap pengujian bertujuan untuk menguji aplikasi, menganalisa hasil pengujian, serta penarikan kesimpulan, dan saran.

3.1 Analisa

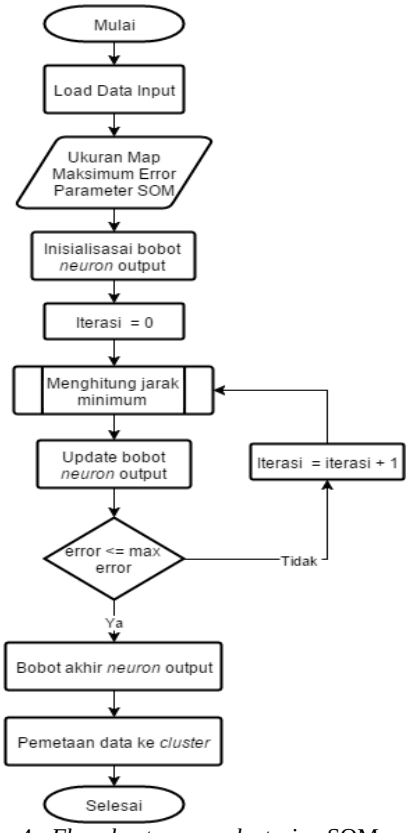
Pada tahap ini dilakukan dengan mengidentifikasi kebutuhan sistem *clustering* mahasiswa pada matakuliah proyek menggunakan algoritma *Self Organizing Map* (SOM). Berikut adalah hasil analisa kebutuhan fungsional sistem :

Tabel 1 : Kebutuhan fungsional sistem

No	Kebutuhan Fungsional Sistem	ID
1	Sistem dapat melakukan setting variabel untuk proses *clustering*	Req-01
2	Sistem dapat membaca data input berupa file csv	Req-02
3	Sistem dapat menyimpan data input dalam *memory*	Req-03
4	Sistem dapat menginisialisasi bobot *neuron* secara acak	Req-04
5	Sistem dapat melakukan *training* data menggunakan algoritma SOM	Req-05
6	Sistem dapat menampilkan data sesuai *cluster*nya	Req-06
7	Sistem dapat menampilkan grafik korelasi antar matakuliah	Req-07
8	Sistem dapat menampilkan tingkat kesalahan hasil *clustering*	Req-08
9	Sistem dapat mengekspor hasil	Req-09

clustering dalam format excel

Kebutuhan sistem ini secara umum mengelola data input, proses *clustering*, dan pemetaan *cluster*. Untuk lebih jelasnya mengenai alur proses*clustering* menggunakan algoritma SOM digambarkan pada *flowhcart* berikut :



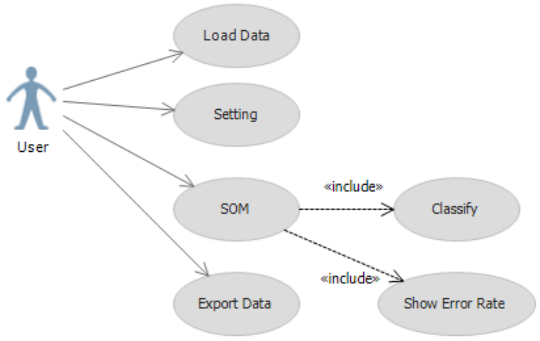
Gambar 4 : *Flowchart* proses *clustering* SOM

Berdasarkan *flowchart* proses *clustering* SOM pada Gambar 4, berikut adalah uraian prosesnya :

1. Memuat data input berupa data nilai mahasiswa sebagai data *training*.
2. Menentukan ukuran map (x, y) sebagai jumlah *cluster*, maksimum *error* yang diharapkan, dan parameter SOM.
3. Inisialisasikan bobot *neuron* output secara acakdengan rentang minimum dan maksimum sesuai bobot di data *training*nya.
4. Set iterasi = 0, sebagai iterasi awal untuk memulai proses *training* data.
5. Menghitung jarak minimum. Data input akan dipilih secara acak kemudian akan dihitung jaraknya dengan masing-masing *neuron* output sampai menemukan *winning neuron* (*neuron* dengan jarak terkecil). Rumus yang digunakan adalah *euclidian distance*.
6. Mengupdate nilai bobot *neuron* output yang terkoneksi dengan *winning neuron*.
7. Kondisi apakah *error* saat ini lebih kecil dari *error* maksimum. Jika Ya, maka lanjut ke langkah 8, jika tidak maka kembali ke langkah 5.
8. Bobot akhir *neuron* yang didapatkan pada masing-masing *cluster* setelah proses *training*.
9. Memetakan data input kedalam *cluster*.Data input akan dihitung jaraknya denganmasing-masing *neuron* output menggunakan rumus *euclidian distance*.*Neuron* output dengan jarak yang terkecil akan menjadi*cluster*-nya.

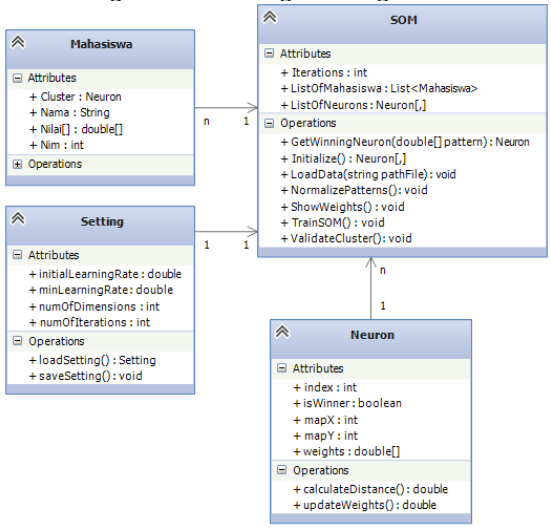
3.2 Perancangan dan Implementasi

Setelah tahap analisa selesai dilakukan, selanjutnya adalah perancangan sistem. Kebutuhan sistem yang telah didefinisikan sebelumnya pada Tabel 1, kemudian dimodelkan dalam bentuk diagram *use case*. Diagram ini menggambarkan fungsionalitas utama sistem yang dapat diakses oleh user.



Gambar 5 : Diagram *use case* sistem

Kebutuhan data pada sistem dimodelkan dalam bentuk diagram kelas sebagaimana gambar berikut :



Gambar 6 : Diagram kelas sistem

Terdapat 4 kelas yang dirancang untuk mengelola data pada sistem *clustering* ini, yaitu kelas *SOM*,

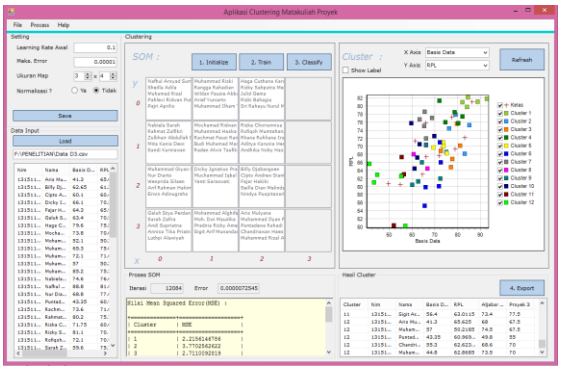
Neuron, *Mahasiswa*, dan *Setting*. Kelas *SOM* digunakan untuk mengelola proses *clustering*, kelas *Mahasiswa* digunakan untuk mengelola data input, kelas *Neuron* digunakan untuk mengelola data *neuron* output, dan kelas *Setting* digunakan untuk mengelola data parameter SOM. Hasil rancangan ini kemudian diimplementasikan menggunakan bahasa C#.

3.3 Pengujian

Pengujian dilakukan denganmenguji fungsionalitas sistem, dan mengevaluasi *cluster* yang dihasilkannya. Pengujian fungsionalitas sistem dilakukan secara *blackbox*, sedangkan untuk mengevaluasi hasil *cluster* dilakukan dengan mengukur nilai *mean square error* (MSE) terhadapbeberapapercobaan *clustering*menggunakan data yang sama. Hasil ini kemudian dianalisa untuk menarik kesimpulan dari penelitian ini.

4. HASIL DAN PEMBAHASAN

Sistem *clustering*telah dikembangkan sesuai kebutuhan. Data input berupa file CSV (*Comma Separated Value*) yang berisi data mahasiswa dan data nilai matakuliah. Format file inputterdiri dari: nim, nama, nilai matakuliah1, nilai matakuliah2, ... nilai matakuliahN, dst. Banyaknya dimensi input tergantung dari jumlah matakuliah yang akan dijadikan dasar *clustering*. Selama proses *training*berlangsung sistem akan menampilkan *error* untuk setiap iterasinya, danakan berhenti ketika telah mencapai*error* terkecil. Nilai akhir dari bobot *neuron* output akan dijadikan acuan dalam proses pemetaan data input kedalam*cluster*serta*plotting*data pada *chart* sesuai *cluster*-nya.Sistem akan menampilkan nilai *mean square error* (MSE) dari masing-masing *cluster* untuk dievaluasi, semakin kecil nilai MSE menunjukan *cluster* tersebut semakin konvergen. Tampilan sistem yang telah dikembangkandapat dilihat pada Gambar 7:



Gambar 7: Sistem*Clustering* Matakuliah Proyek

4.1 Pengujian Sistem

Sistem yang telah dikembangkankemudian diuji secara *blackbox* untuk mengetahui apakah fungsionalitas sistem telah berjalan dengan baik atau tidak.Masing-masing modul sistem yang telah diimplementasikan selanjutnya diuji dengan butir uji pada Tabel 2. Hasil pengujian menunjukan bahwa fungsionalitas sistem berjalan dengan baik.

Tabel2 :PengujianBlackBox

Modul	Respon sistem yang diharapkan	Pass / Fail
1. Load Data		
LoadData	Membaca file CSV	*Pass*
	Menyimpan data ke *memory*, dan menampilkannya	*Pass*
2. Setting		
LoadSetting	Memuat data *setting*	*Pass*
SaveSetting	Menyimpan data *setting*	*Pass*
3. SOM		
Initialize	Menginisialisasi nilai bobot *neuron*	*Pass*
TrainSOM	Melakukan *training* data	*Pass*
GetWinning Neuron	Menemukan *wining neuron*	*Pass*
UpdateWeight	Mengupdate nilai bobot *neuron*	*Pass*
ShowWeight	Menampilkan bobot *neuron*	*Pass*
Classify	Memetakan data input ke *cluster*	*Pass*
4. Export Data		
ExportData	Menghasilkan file excel sesuai hasil *cluster*	*Pass*

4.2 Pengujian *Clustering*

Untuk mengevaluasi hasil *clustering* sistem, pengujian dilakukandengan mengambil data sampel, yaitu data nilai semester ganjil mahasiswa D3 kelas 2A dan 2B tahun ajaran 2014/2015. Data sampel berjumlah57 data, dan akan di-*cluster* sebanyak 12 *cluster*. Parameter-parameter padaujicobaini, yaitu : ukuran map (3x4);*learning rate* 0,1; dan maksimum *error* 0,00001. Berikutini adalahhasil*clustering*sistemnya:

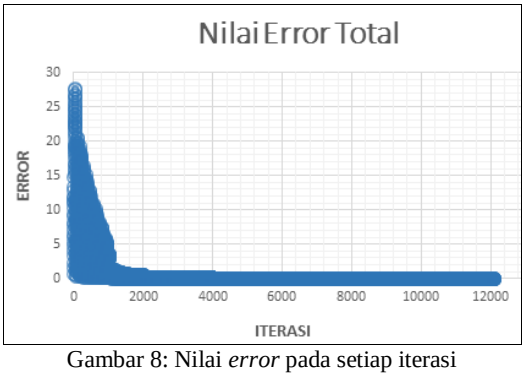
Tabel3 :Hasil clustering sistem

Cluster	Anggota	Nilai MSE
1	M15,M29,M43, M45,M46	2,2156
2	M13,M49, M54, M56,M57	3,7702

3	M07,M21,M38, M50,M53	2,7110
4	M14,M19,M26 M40,M51	2,4440
5	M08,M11,M18 M35,M48	2,1875
6	M20,M22,M28 M30,M31	4,2843
7	M10,M16,M25 M34,M37	3,1543
8	M04,M42,M55	3,0240
9	M02,M03,M05, M24, M27	3,4658
10	M06,M23,M32 M33,M39	3,7441
11	M09,M41,M47, M52	4,4960
12	M01,M12,M17, M36,M44	6,2514
Rata-rata MSE		3,4790

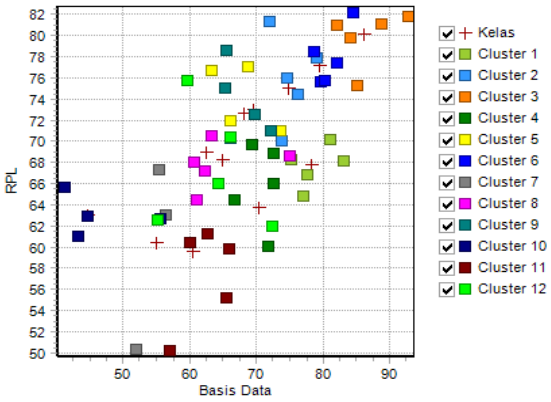
Tabel 3 menunjukan hasil *clustering* sistem, dimana secara umum masing-masing *cluster* memiliki 5 anggota dengan nilai MSE yang berbeda-beda. *Cluster* terbaik adalah *cluster* dengan nilai MSE yang paling kecil, yaitu *cluster* 5 dengan nilai 2,1875; sedangkan *cluster* 12 adalah *cluster* yang terburuk dengan nilai MSE 6,2514. Dari hasil percobaan tersebut rata-rata MSE dari *cluster* yang terbentuk adalah 3,4790.

Untuk mencapai *error* terkecil yang diharapkan (1 x 10^{-5}), proses *training* data pada sistem dilakukan hingga 12084 iterasi. Dimana *error* terbesar pada iterasi ke 57 dengan nilai *error*27,3097.Penurunan *error* terlihat signifikan sampai iterasi ke 1254, dan untuk iterasi selanjutnya *error* menurun secara konstan. Proses *training*data berhenti pada iterasi ke 12084 dengan nilai *error*7,0874 x 10^{-6}. Berikut adalah nilai *error* yang dicapai pada setiap iterasinya :

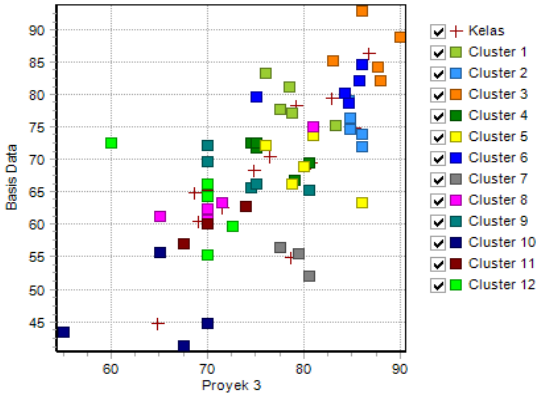


Gambar 8: Nilai *error* pada setiap iterasi

Matakuliah yang menjadi parameter pada proses *clustering* ini, yaitu : Basis Data, RPL, Aljabar Linear, dan Proyek 3. Untuk melihat kedekatan nilai pada masing-masing anggota *cluster*-nya, data digambarkan dalam bentuk grafik 2 dimensi dengan *x-axis* dan *y-axis* berupa matakuliah-matakuliah tersebut. Pada Gambar 9, setiap anggota *cluster* cenderung memiliki kedekatan nilai pada matakuliah RPL, dan Basis Data. Begitu juga pada Gambar 10,setiap anggota *cluster* cenderung memiliki kedekatan nilai pada pada matakuliah Basis Data, dan Proyek 3. Hal ini menunjukan bahwa algoritma SOM akan mengelompokkan data berdasarkan kedekatan nilai di matakuliah-matakuliah tersebut, sehingga masing-masing *cluster* memiliki anggota yang konvergen.



Gambar 9: Kedekatan nilai RPL dan Basis Data



Gambar 10: Kedekatan nilai Basis Data dan Proyek3

Untuk mengukur rata-rata nilai MSE hasil *clustering* algoritma SOM, pengujian dilakukan sebanyak 5 kali dengan data danskenario pengujian yang sama. Hasil pengujian dapat dilihat pada Tabel 4 berikut :

Tabel 4 : Rata-rata MSE pada 5 kali pengujian

Pengujian ke -	Rata-rata MSE
1	3,4790
2	3,2802
3	3,5344
4	3,5406
5	3,3721

Rata-rata MSE pada setiap pengujian memiliki nilai yang berbeda. Perbedaan nilai ini akibat proses inisialisasi bobot *neuron* secara acak saat awal proses *training* data. Nilai rata-rata MSE yang terkecil adalah 3,2802 pada pengujian ke- 2, sedangkan yang terbesar adalah 3,5406 pada pengujian ke- 4. Namun secara umum, hasilnya tidak jauh berbeda antara satu pengujian dengan pengujian lainnya dengan selisih nilai rata-rata MSE yang terbesar dan terkecil adalah 0,2604.

5. KESIMPULAN

Penelitian ini bertujuan mengimplementasikan algoritma *Self Organizing Map* (SOM) untuk *clustering* mahasiswa pada matakuliah proyek, dan mengevaluasi hasil *clustering* sistemnya. Parameter yang dijadikan dasar *clustering* adalah nilai-nilai matakuliah di semester berjalan.Fungsionalitassistem yang telahdikembangkanberjalandenganbaik, demikian pula denganhasil*clustering*sistemnya. Dari beberapapercobaantidakterlihatperbedaannilai*mean square error* (MSE) yang signifikan, nilaiMSE yang terbesaradalah 3,5406 danterkecil3,2802 denganselisihnilai 0,2604. Hal inimenunjukan*cluster* yang terbentukolehalgoritmaSOM padastudikasusinibersifatkonvergen. Perbedaannilai MSE padaalgoritma SOM merupakanakibatinisialisasi bobot *neuron* secara acak saat awal proses *training* data.Topikpenelitianselanjutnyadapatmenelitibagaimana agar nilai MSE inidapatdikurangi agar *cluster* yang dihasilkanlebihbaik.

UCAPAN TERIMAKASIH

Ucapan terimakasih disampaikan kepada segenap sivitas akademika Jurusan Teknik Komputer dan Informatika Poliktenik Negeri Bandung yang telah berkenan membantu pengumpulan data selama penelitian berlangsung, serta segenap staf UPPM Politeknik Negeri Bandung yang telah memberikan kesempatan untuk melaksanakan penelitian dosen Pemula tahun 2015. Ucapan terimakasih juga disampaikan kepada reviewer penelitian ini, IbuTransmissiaSemiawan, Ph.D, dan Bapak Drs. Sardjito, M.Scatasmasukan yang diberikan dalammenyelesaikanpenelitianini.

DAFTAR PUSTAKA

[1] Gan, Guojun. *Data Clustering in C++, An Object-Oriented Approach*. *Chapter 1 (Data Clustering)*. Chapman & Hall. USA. 2011.

[2] Kohonen, Teuvo. *The Self-Organizing Map*. Proceeding of IEEE, Vol 78, No 9, September 1990.

[3] Maria, F, dkk. *Using Self Organizing Maps in Applied Geomorphology. Self Organizing Maps - Applications and Novel Algorithm Design*. InTech. Croatia. 2011.

[4] Lestari, Wiji. Sistem *Clustering* Kecerdasan Majemuk Mahasiswa Menggunakan Algoritma *Self Organizing Map* (SOM). STMIK Duta Bangsa. Surakarta. 2011.

[5] Sote, A.M, Pande, S.R. *Web Page Clustering Using Self-Organizing Map*. International Journal of Computer Science and Mobile Computing. India. 2015.

[6] Dwi, Andharini Cahyani, dkk. Perbandingan Metode SOM (*Self Organizing Map*) Dengan Pembobotan Berbasis RBF (*Radial Basis Function*). Jurnal Teknologi Technoscientia. 2014.

[7] Object Management Group. *What is UML*. http://www.omg.org/gettingstarted/what_is _uml.htm. Diakses pada 10 Maret. 2015.

[8] Williams, Laurie. *An Introduction to the Unified Modelling Language*. http://agile.csc.ncsu.edu/ SEMaterials/UMLOverview.pdf. Diakses pada 5 Maret. 2015.

[9] Deitel, Paul, Deitel, Harvey. *C# 2010 for Programmers Fourth Edition*. Prentice Hall. USA. 2011.

[10] TechNet Microsoft. https://technet. microsoft.com/ en-us/library/bb496996.aspx. Diakses pada 10 September. 2015.

[11] Puspha, C. N, dkk. *Web Page Recommendation System Using Self Organizing Map Technique*. International Journal of Current Engineering and Technology. India. 2014.

Program Linier (LP) adalah suatu teknik yang sangat banyak digunakan dalam optimisasi. Dalam aplikasinya suatu model LP banyak mengkaitkan atau menggunakan parameter yang nilainya diselesaikan oleh ahli/pembuat keputusan.Bagaimanapun keduanya (ahli/pembuat keputusan) tidak mengetahui nilai dari paramater kebanyakan kasus. Oleh karena itu program linier fuzzy telah dikenalkan dan dipelajari. Pandian [1] telah menunjukan atau mengenalkan suatu pendekatan terbaru yang dinamakan metode *sum of objectives* (SO) untuk menemukan solusi efisien untuk menyelesaikan masalah program linier *multi-objective*.

Dalam penelitian ini akan dibahas metode baru yaitu metode *Level-Sum* untuk menemukan solusi optimal *fuzzy* untuk masalah program linier *fuzzy*. Dengan menggunakan contoh numerik, metode *level-sum*

dapat diilustrasikan. Keuntungan dari metode yang diusulkan adalah fungsi peringkat *fuzzy* yang tidak digunakan, hasil dapat dilakukan dengan penyelesaian LP sehingga memperoleh semua kendala dan perhitungan karena hanya didasarkan pada teknik *crisp* LP.

Penggunaan Metode Level Sum ini, dapat membantu ahli atau pembuat keputusan untuk mendapatkan solusi optimum meskipun nilai parameter tidak diketahui dengan pasti.

2. KAJIAN LITERATUR

2.1 Fungsi Linier Fuzzy

Program linier *fuzzy* adalah program linier yang dinyatakan dengan fungsi objektif dan fungsi kendala yang memiliki parameter *fuzzy* dan ketidaksamaan *fuzzy*[2]. Tujuan dari program linier *fuzzy* adalah mencari suatu nilai z yang merupakan fungsi objektif yang akan dioptimasikan sedemikian sehingga tunduk pada batasan-batasan yang dimodelkan dengan menggunakan himpunan *fuzzy*[3].

Masalah program linier yang keseluruhan variabel dan koefisien-koefisien yang digunakan berbentuk data *fuzzy*, serta operasi-operasi aritmatik yang digunakan adalah operasi aritmatik bilangan fuzzy disebut masalah program linier *fully fuzzy* (FFLP). Keunggulan FFLPdibandingkan program linier adalah adanya penjelasan tentang koefisien biaya, variabel, maupun koefisien teknis yang tidak bernilai tegas dalam kehidupan nyata yang dapat digambarkan dengan menggunakan bilangan *fuzzy* [2].

2.2 Solusi Optimal Fuzzy

Adapun langkah-langkah yang dilakukan dalam menentukan solusi optimal pada permasalahan program linier berparameter *fuzzy* adalah sebagai berikut [4] :

1. Merumuskan permasalahan yang mengandung parameter *fuzzy* dalam bentuk program linier *fuzzy*, yang terdiri dari :
   a. Program linier dengan parameter *fuzzy* pada fungsi pembatas $\tilde{b}_i$
   b. Program linier yang berparameter *fuzzy* pada fungsi objektif
2. Menentukan fungsi keanggotaan dari himpunan *fuzzy*
3. Maksimasi $Z = \sum_{i=1}^{n} c_i x_i$ terhadap fungsi pembatas $\sum_{j=1}^{n} a_{ij} x_j \lesssim \tilde{b}_i$ dan fungsi objektif
4. Mendefinisikan fungsi keanggotaan yang menggambarkan derajat optimalitas dari setiap fungsi objektif $Z = \sum_{i=1}^{n} c_i x_i$
5. Menentukan solusi optimal dengan menggunakan metode simpleks

Dalam penelitian ini telah dibuktikan teorema yang memperlihatkan suatu hubungan antara solusi optimal *fuzzy* untuk masalah program linier *fuzzy* (P) dan solusi efisien untuk masalah program linier *multi-objective* (M) [1].

Teorema1

Misal $X^* = \{x^*_j, y^*_j, t^*_j; j = 1,2,\dots,m\}$ adalah solusi efisien untuk masalah(M), maka $\tilde{X}^* = \{(x^*_j, y^*_j, t^*_j); j = 1,2,\dots,m\}$ adalah solusi layak untuk masalah (P).

Asumsikan bahwa $\tilde{X}^* = \{(x^*_j, y^*_j, t^*_j); j = 1,2,\dots,m\}$ tidak optimal untuk malasah (P). Maka, terdapat solusi layak $\tilde{X} = \{(x_j, y_j, t_j); j = 1,2,\dots,m\}$ untuk masalah (P) sehingga $Z(\tilde{X}) \succ Z(\tilde{X}^*)$ dimana $z_i(x,y,t) \geq z_i(x^*,y^*,t^*), i = 1,2,3$ dan $z_r(x,y,t) > z_r(x^*,y^*,t^*),\ untuk\ beberapa\ r \in \{1,2,3\}$

dimana $x^* = \{x^*_j;\ j = 1,2,\dots,m\}$, $y^* = \{y^*_j;\ j = 1,2,\dots,m\}$, $t^* = \{t^*_j;\ j = 1,2,\dots,m\}$, $x = \{x_j;\ j = 1,2,\dots,m\}$, $y = \{y_j;\ j = 1,2,\dots,m\}$, dan $t = \{t_j;\ j = 1,2,\dots,m\}$. Ini berarti bahwa $\tilde{X}^* = \{(x^*_j, y^*_j, t^*_j); j = 1,2,\dots,m\}$ bukan solusi efisien untuk masalah (M) dimana terjadi kontradiksi. Sehingga teorema ini terbukti.

2.3 Program Linier Multi Objektif

Program linier *multi objective* (MOLP)adalah metode optimasi dengan beberapa fungsi tujuan yang tunduk pada beberapa batasan. Solusi permasalahan ini diperoleh seperti penyelesaian optimasi dengan 1 fungsi tujuan [3].

Berdasarkan masalah optimasi *multi-objective* sebagai berikut [10]:

(MP) Minimasi $f(x) = (f_1(x), f_2(x), \dots, f_k(x))$

Kendala $g(x) \geqq 0, x \in X$

Dimana $f_i: X \longrightarrow R, i = 1,2,\dots,k$ dan $g: X \longrightarrow R^m$, dimana $g = (g_1, g_2, \dots, g_m$ adalah fungsi terdiferensiasi di X , sebuah subset *convex* terbuka dariR^n.

Sekarang $P = \{x \in X, g_j(x) \leq 0, j = 1,2,\dots,m\}$ adalah himpunan semua solusi yang layak untuk masalah tersebut.

Definisi 2
Titik layak x^* dikatakan efisien untuk masalah program *multi-objective* jika terdapat titik layak x di P sehingga $f_i(x) \leq f_i(x^*), i = 1,2,\dots,k$ dan $f_r(x) < f_r(x^*)$, dimana $r \in \{1,2,\dots,k\}$

3.HASIL DAN PEMBAHASAN

3.1 Pencarian Solusi Optimal *Fuzzy* untuk Masalah Program Linier *Fuzzy* Menggunakan Metode *Level-Sum*

3.1.1 Masalah Program Linier *Fully Fuzzy*

Diperlukan definisi dari operasi dasar aritmatika hubungan orde parsial dari bilangan *triangularfuzzy* berdasarkan dari fungsi dasar yang dapat ditemukan di [4,5,6].

Definisi 3
Bilangan *fuzzy* $\tilde{a}$ adalah bilangan triangular *fuzzy* ditunjukkan oleh (a_1, a_2, a_3) dimana $a_1, a_2,$ dan a_3 adalah bilangan asli dan fungsi anggotanya $\mu_{\tilde{a}}(x)$ diberikan di bawah ini :

$$\mu_{\tilde{a}}(x) = \begin{cases} \frac{x-a_1}{a_2-a_1}, untuk\ a_1 \leq x \leq a_2 \\ \frac{a_3-x}{a_3-a_2}, untuk\ a_2 \leq x \leq a_3 \\ 0 \quad , untuk\ lainnya \end{cases}$$

Misalkan F (R) adalah himpunan semua bilangan nyata *triangularfuzzy*.

Definisi 4
Misal (a_1, a_2, a_3) dan (b_1, b_2, b_3) ada di F (R). Maka,

(i) $(a_1, a_2, a_3) \oplus (b_1, b_2, b_3) = (a_1 + b_1, a_2 + b_2, a_3 + b_3)$

(ii) $(a_1, a_2, a_3) \ominus (b_1, b_2, b_3) = (a_1 - b_1, a_2 - b_2, a_3 - b_3)$

(iii) $k(a_1, a_2, a_3) = (ka_1, ka_2, ka_3), untuk\ k \geq 0$

(iv) $k(a_1, a_2, a_3) = (ka_3, ka_2, ka_1), untuk\ k < 0$

(v) $(a_1, a_2, a_3) \otimes (b_1, b_2, b_3) =$

$$\begin{cases} (a_1 b_1, a_2 b_2, a_3 b_3), & a_1 \geq 0 \\ (a_1 b_3, a_2 b_2, a_3 b_3), & a_1 < 0, a_3 \geq 0 \\ (a_1 b_1, a_2 b_2, a_1 b_3), & a_3 < 0 \end{cases}$$

Berdasarkan program linier *fully fuzzy* dengan *fuzzy* kendala ketidaksamaan/persamaan dan variabel *fuzzy* yang dapat diformulasikan sebagai berikut [1]:

(P) Maksimasi $\tilde{z} \approx \tilde{c}^T \tilde{x}$

Terhadap kendala $\tilde{A} \otimes \tilde{x} \{\preceq, \approx, \succeq\} \tilde{b}, \tilde{x} \succeq \tilde{0}$

Dimana $\tilde{a}_{ij}, \tilde{c}_j, \tilde{x}_j, \tilde{b}_i \in F(R)$ untuk semua $1 \leq j \leq n$ dan $1 \leq i \leq m, \tilde{c}^T = (\tilde{c}_j)_{1\times n}$,$\tilde{A} = (\tilde{a}_{ij})_{m\times n}$, $\tilde{x} = (\tilde{x}_j)_{n\times 1}$ dan $\tilde{b} = (\tilde{b}_i)_{m\times 1}$

Misal parameter $\tilde{z}, \tilde{a}_{ij}, \tilde{c}_j, \tilde{x}_j$ dan $\tilde{b}_i$ adalah bilangan*triangular fuzzy* $(z_1, z_2, z_3), (p_j, q_j, r_j), (x_j, y_j, t_j), (a_{ij}, b_{ij}, c_{ij})$ dan (b_i, g_i, h_i) berturut-turut. Kemudian, masalah (P) dapat dituliskan sebagai berikut :

(P) Maksimasi $(z_1, z_2, z_3) \approx \sum_{n=1}^{j} (p_j, q_j, r_j) \otimes (x_j, y_j, t_j)$

Terhadap kendala

$\sum_{n=1}^{j} (a_{ij}, b_{ij}, c_{ij}) \otimes (x_j, y_j, t_j) \{\preceq, \approx, \succeq\} (b_i, g_i, h_i)$untuk semua $i = 1,2,\dots,m. (x_j, y_j, t_j) \succeq \tilde{0}, j = 1,2,\dots,m$

Dengan menggunakan operasi aritmatika dan relasi orde parsial, masalah program linier *fuzzy*dapat dituliskan sebagai masalah program linier *multi-objective* yang diberikan sebagai berikut :

(M)Maksimasi

$z_1 = \sum_{n=1}^{j} nilai\ bawah\ dari\ \left((p_j, q_j, r_j) \otimes (x_j, y_j, t_j)\right)$

Maksimasi

$z_2 = \sum_{n=1}^{j} nilai\ tengah\ dari\ \left((p_j, q_j, r_j) \otimes (x_j, y_j, t_j)\right)$

Maksimasi

$z_1 = \sum_{n=1}^{j} nilai\ atas\ dari\ \left((p_j, q_j, r_j) \otimes (x_j, y_j, t_j)\right)$

Terhadap kendala

$\sum_{n=1}^{j} nilai\ bawah\ dari\ \left((a_{ij}, b_{ij}, c_{ij}) \otimes (x_j, y_j, t_j)\right) \{\leq, =, \geq\} b_i$

, untuk semua $i = 1,2,\dots,m;$

$\sum_{n=1}^{j} nilai\ tengah\ dari\ \left((a_{ij}, b_{ij}, c_{ij}) \otimes (x_j, y_j, t_j)\right)\{\leq, =, \geq\} g_i$

, untuk semua $i = 1,2,\dots,m;$

$\sum_{n=1}^{j} nilai\ atas\ dari\ \left((a_{ij}, b_{ij}, c_{ij}) \otimes (x_j, y_j, t_j)\right)\{\leq, =, \geq\} h_i$

, untuk semua $i = 1,2,\dots,m;$

$z_2 \geq z_1\ ;\ z_3 \geq z_2\ ;\ x_j \leq y_j, j = 1,2,\dots,m\ ;\ y_j \leq t_j, j = 1,2,\dots,m\ ;\ x_j \geq 0,$
$j = 1,2,\dots,m$

3.1.2 Metode *Level-Sum*

Metode *level-sum* merupakan sebuah metode baru untuk mencari solusi optimal *fuzzy* untuk masalah program linier *fully fuzzy* yang didasarkan pada program linier *multi-objective* dan metode simpleks.

Adapun langkah-langkah dalam metode ini adalah sebagai berikut :

1. Buat sebuah *crisp* masalah program linier *multi-objective* dari masalah program linier *fuzzy* yang diberikan.
2. Tentukan suatu solusi efisien untuk masalah program linier *multi-objective* berdasarkan pada langkah 1. Dengan menggunkan metode *sum-objective*[7].
3. Solusi efisien yang didapatkan dari langkah 2 untuk masalah program linier *multi-objective* menghasilkan suatu solusi optimal *fuzzy* untuk masalah program linier *fuzzy*.

3.2 Contoh Numerik

Metode *level-sum* dapat di ilustrasikan dengan contoh numerik, dimana pada contoh 1 menggunakan masalah program linier *fully fuzzy* yang keseluruhannya menggunakan *fuzzy*, baik itu bilangan maupun operasi bilangannya. Berbeda dengan contoh 2 yang menggunakan masalah program linier *fuzzy* dimana operasinya saja yang menggunakan operasi bilangan *fuzzy*. Untuk lebih jelasnya dapat dilihat pada contoh berikut :

Contoh 1

Berdasarkan masalah program linier *fully fuzzy* berikut :

Maksimumkan $\tilde{z} \approx (-1,2,3) \otimes \tilde{x}_1 \oplus (2,3,4) \otimes \tilde{x}_2$

Terhadap kendala

$(0,1,2) \otimes \tilde{x}_1 \oplus (1,2,3) \otimes \tilde{x}_2 \approx (2,10,24);$

$(1,2,3) \otimes \tilde{x}_1 \oplus (0,1,2) \otimes \tilde{x}_2 \approx (1,8,21);$

$\tilde{x}_1, \tilde{x}_2 \geq 0$

Penyelesaian:

Misal $\tilde{x}_1 = (x_1, y_1, t_1)$ dan $\tilde{x}_2 = (x_2, y_2, t_2)$

Gunakan langkah pertama untuk mengubah masalah program linier *fully fuzzy* menjadi masalah program linier *multi-objective*, sehingga menjadi :

(M) Maksimumkan $z_1 = -t_1 + 2x_2$

$z_2 = 2y_1 + 3y_2$

$z_3 = 3t_1 + 4t_2$

Terhadap kendala

$x_2 = 2\ ;\ y_1 + 2y_2 = 10\ ;\ 2t_1 + 3t_2 = 24\ ;\ x_1 = 1\ ;\ 2y_1 + y_2 = 8\ ;3t_1 + 2t_2 = 21\ ;\ z_2 \geq z_1\ ;\ z_3 \geq z_2\ ;\ y_1 \geq x_1\ ;\ y_2 \geq x_2\ ;\ t_1 \geq y_1\ ;\ t_2 \geq y_2\ ;\ x_1, x_2 \geq 0$

Dengan menggunakan langkah kedua maka akan diperoleh :

(S)

Maksimumkan $z = 2x_2 + 2y_1 + 3y_2 + 2t_1 + 4t_2$

Terhadap kendala

$x_2 = 2\ ;\ y_1 + 2y_2 = 10\ ;\ 2t_1 + 3t_2 = 24\ ;\ x_1 = 1\ ;\ 2y_1 + y_2 = 8\ ;3t_1 + 2t_2 = 21\ ;\ -2x_2 + 2y_1 + 3y_2 + t_1 \geq 0\ ;\ -2y_1 - 3y_2 + 3t_1 + 4t_2 \geq 0\ ;\ y_1 - x_1 \geq 0\ ;\ y_2 - x_2 \geq 0\ ;\ t_1 - y_1 \geq 0\ ;\ t_2 - y_2 \geq 0;\ x_1, x_2 \geq 0$

Selesaikan masalah (S) dengan menggunakan metode simpleks, dimana dalam proses penyelesaian metode simpleks ini dibantu dengan menggunakan sebuah aplikasi yang bernama POM (*Production and Operation Management*). Sehingga akan didapatkan solusi optimal untuk masalah (S) adalah $x_1 = 1\ ;\ x_2 = 2\ ;\ y_1 = 2\ ;\ y_2 = 4\ ;\ t_1 = 3$ dan $t_2 = 6$ dengan $Z = 50$. Jadi, $(x_1 = 1\ ;\ x_2 = 2\ ;\ y_1 = 2\ ;\ y_2 = 4\ ;\ t_1 = 3\ ;\ t_2 = 6)$ adalah solusi efisien untuk masalah (M). Dengan menggunakan langkah ketiga akan diperoleh $\tilde{x}_1 \approx (1,2,3)$ dan $\tilde{x}_2 \approx (2,4,6)$ dan $\tilde{z} \approx (1,16,33)$ adalah solusi optimal *fuzzy* untuk masalah program linier *fully fuzzy*.

Contoh 2

Berdasarkan masalah program linier *fuzzy* berikut:

Maksimumkan

$\tilde{z} \approx (7,10,14,25) \otimes x_1 \oplus (20,25,35,40) \otimes x_2$

Terhadap kendala

$(1,3,4) \otimes x_1 \oplus (2,6,7) \otimes x_2 \preceq (8,13,15);$

$(3,4,6) \otimes x_1 \oplus (1,6,10) \otimes x_2 \preceq (3,7,9); x_1, x_2 \geq 0$

Penyelesaian :

Misal $\tilde{z}_1 = (z_1, z_2, z_3, z_4)$

Gunakan langkah pertama untuk mengubah masalah program linier *fuzzy* menjadi masalah program linier *multi-objective*, sehingga menjadi:

(M) Maksimumkan$z_1 = 7x_1 + 20x_2$
Maksimumkan$z_2 = 10x_1 + 25x_2$
Maksimumkan$z_3 = 14x_1 + 35x_2$
Maksimumkan$z_4 = 25x_1 + 40x_2$
Terhadap kendala

$x_1 + 2x_2 \leq 8;\ 3x_1 + 6x_2 \leq 13;\ 4x_1 + 7x_2 \leq 15;\ 3x_1 + x_2 \leq 3;\ 4x_1 + 6x_2 \leq 7;\ 6x_1 + 10x_2 \leq 9;\ z_2 \geq z_1;\ z_3 \geq z_2;\ z_4 \geq z_3; x_1, x_2 \geq 0$

Dengan menggunakan langkah kedua maka akan diperoleh :

(S) Maksimumkan $z = 56x_1 + 120x_2$
Terhadap kendala

$x_1 + 2x_2 \leq 8;\ 3x_1 + 6x_2 \leq 13;\ 4x_1 + 7x_2 \leq 15;\ 3x_1 + x_2 \leq 3;\ 4x_1 + 6x_2 \leq 7;\ 6x_1 + 10x_2 \leq 9;\ 3x_1 + 5x_2 \geq 0;\ 4x_1 + 10x_2 \geq 0;\ 11x_1 + 5x_2 \geq 0;\ x_1, x_2 \geq 0$

Selesaikan masalah (S) dengan menggunakan metode simpleks, dimana dalam proses penyelesaian metode simpleks ini dibantu dengan menggunakan sebuah aplikasi yang bernama POM (*Production and Operation Management*). Sehingga akan didapatkan solusi optimal untuk masalah (S) adalah $x_1 = 0$ dan $x_2 = 0.9$ dengan $Z = 108$. Jadi, $(x_1 = 0, x_2 = 0.9)$ adalah solusi efisien untuk masalah (M).

Dengan menggunakan langkah ketiga akan diperoleh $x_1 = 0, x_2 = 0.9$ dan $\tilde{z} \approx (18, 25.5, 31.5, 36)$ adalah

solusi optimal *fuzzy* untuk masalah program linier *fuzzy*.

4. KESIMPULAN

Adapun proses menyelesaikan masalah program linier *fuzzy* dengan menggunakan metode *level-sum* sehingga diperoleh penyelesaian optimal *fuzzy* adalah sebagai berikut :

1. Buat sebuah *crisp* masalah program linier *multi-objective* dari masalah program linier *fuzzy* yang diberikan.
   Misal parameter $\tilde{z}, \tilde{a}_{ij}, \tilde{c}_j, \tilde{x}_j$ dan $\tilde{b}_i$ adalah bilangan *triangular fuzzy* $(z_1, z_2, z_3), (p_j, q_j, r_j), (x_j, y_j, t_j), (a_{ij}, b_{ij}, c_{ij})$ dan (b_i, g_i, h_i) berturut-turut. Kemudian, masalah (P) dapat dituliskan sebagai berikut :

(P) Maksimasi
$(z_1, z_2, z_3) \approx \sum_{n=1}^{j} (p_j, q_j, r_j) \otimes (x_j, y_j, t_j)$
Terhadap kendala

$\sum_{n=1}^{j} (a_{ij}, b_{ij}, c_{ij}) \otimes (x_j, y_j, t_j) \{\leq, \approx, \geq\} (b_i, g_i, h_i)$
untuk semua $i = 1,2,\dots,m. (x_j, y_j, t_j) \geq \tilde{0}, j = 1,2,\dots,m$
Dengan menggunakan operasi aritmatika dan relasi orde parsial, masalah program linier *fuzzy* dapat dituliskan sebagai masalah program linier *multi-objective* yang diberikan sebagai berikut :
(M) Maksimasi

$z_1 = \sum_{n=1}^{j} nilai\ bawah\ dari\ \left((p_j, q_j, r_j) \otimes (x_j, y_j, t_j)\right)$

Maksimasi

$z_2 = \sum_{n=1}^{j} nilai\ tengah\ dari\ \left((p_j, q_j, r_j) \otimes (x_j, y_j, t_j)\right)$

Maksimasi

$z_1 = \sum_{n=1}^{j} nilai\ atas\ dari\ \left((p_j, q_j, r_j) \otimes (x_j, y_j, t_j)\right)$

Terhadap kendala
$\sum_{n=1}^{j} nilai\ bawah\ dari\ \left((a_{ij}, b_{ij}, c_{ij}) \otimes (x_j, y_j, t_j)\right) \{\leq, =, \geq\} b_i$
, untuk semua $i = 1,2,\dots,m;$
$\sum_{n=1}^{j} nilai\ tengah\ dari\ \left((a_{ij}, b_{ij}, c_{ij}) \otimes (x_j, y_j, t_j)\right) \{\leq, =, \geq\} g_i$
, untuk semua $i = 1,2,\dots,m;$
$\sum_{n=1}^{j} nilai\ atas\ dari\ \left((a_{ij}, b_{ij}, c_{ij}) \otimes (x_j, y_j, t_j)\right) \{\leq, =, \geq\} h_i$
, untuk semua $i = 1,2,\dots,m;$
$z_2 \geq z_1;\ z_3 \geq z_2;\ x_j \leq y_j, j = 1,2,\dots,m;\ y_j \leq t_j, j = 1,2,\dots,m;$
$j = 1,2,\dots,m$

2. Tentukan suatu solusi efisien untuk masalah program linier *multi-objective* berdasarkan pada langkah 1. Dengan menggunkan metode *sum-*

objective[7], dan selesaikan dengan menggunakan metode simpleks.

3. Ubah solusi efisien yang didapatkan dari langkah 2 untuk masalah program linier *multi-objective* sehingga menghasilkan suatu solusi optimal *fuzzy* untuk masalah program linier *fuzzy* dengan menggunakan Teorema1.

DAFTAR PUSTAKA

[1] Pandian,P.2013.*Multi-objective Programming Approach for Fuzzy Linear Programming Problems.*Vol. 7, 2013, no. 37, 1811 – 1817.

[2] Otadi.M.2014.*Solving Fully Fuzzy Linear Programming*. Vol 6 No 1, P 19 26.

[3] Kusumadewi, Sri dan Hari Purnomo.2004.Aplikasi Logika *Fuzzy* untuk Pendukung Keputusan.Yogyakarta.Graha Ilmu.

[4] George J. Klir and Bo Yuan.*Fuzzy Sets and Fuzzy logic: Theory and Applications.*Prentice-Hall.2008.

[5] L. A. Zadeh, Fuzzy sets, Information and Control, 8 (1965), 338-353.

[6] R. E. Bellman and L. A. Zadeh, Decision making in a fuzzy environment,Management Science, 17(1970), 141-164.

[7] Pandian, P. 2012. A simple approach for finding a fair solution to multiobjective programming problems. *Bulletin of Mathematical Sciences & Applications*,1(2012), 25 – 30.