

## PENINGKATAN PERFORMA PENGELOMPOKAN SISWA BERDASARKAN AKTIVITAS BELAJAR PADA MEDIA PEMBELAJARAN DIGITAL MENGUNAKAN METODE *ADAPTIVE MOVING SELF-ORGANIZING MAPS*

Onky Prasetyo\*<sup>1</sup>, Ahmad Afif Supianto<sup>2</sup>, Syaiful Anam<sup>3</sup>, Hilman Ferdinandus Pardede<sup>4</sup>, Vicky Zilvan<sup>5</sup>,  
R. Budiarianto Suryo Kusumo<sup>6</sup>

<sup>1,2,3</sup> Universitas Brawijaya, Malang

<sup>2,4,5,6</sup> Pusat Riset Informatika, Badan Riset dan Inovasi Nasional, Jakarta Pusat

Email: <sup>1</sup>railgun48@student.ub.ac.id, <sup>2</sup>ahma083@lipi.go.id, <sup>3</sup>syaiful@ub.ac.id, <sup>4</sup>hilm001@lipi.go.id,  
<sup>5</sup>vick001@lipi.go.id, <sup>6</sup>rbud001@lipi.go.id

(Naskah masuk: 21 September 2021, diterima untuk diterbitkan: 31 Januari 2022)

### Abstrak

Digitalisasi proses pembelajaran memungkinkan untuk dihasilkannya rekaman terhadap setiap aktivitas siswa selama belajar. Rekaman yang dihasilkan tersebut dapat digunakan untuk mengelompokkan siswa berdasarkan pola dari proses belajar yang dilakukan. Hasil pengelompokkan yang diperoleh dapat digunakan untuk melakukan penyesuaian komponen pembelajaran ataupun metode pembelajaran bagi siswa. Salah satu metode pengelompokkan yang sering digunakan adalah *Self-Organizing Maps* (SOM), SOM merupakan metode jaringan syaraf tiruan dengan tujuan untuk mempertahankan topologi data ketika data *input* multidimensi diubah menjadi data *output* dengan dimensi yang lebih rendah. *Neuron* SOM pada dimensi *input* diperbaharui sepanjang proses pelatihan, sedangkan *neuron* pada dimensi *output* tidak mendapatkan pembaruan sama sekali, hal ini menyebabkan struktur *neuron* yang digunakan pada tahapan inisialisasi akan tetap sama hingga akhir proses pengelompokkan. Pada penelitian ini menggunakan metode *Adaptive Moving Self-Organizing Maps* (AMSOM) yang menggunakan struktur *neuron* lebih fleksibel, dengan dimungkinkannya terjadi perpindahan, penambahan dan penghapusan dari *neuron* menggunakan data 12 *assignments* dari media pembelajaran MONSAKUN. Hasil penelitian menunjukkan terdapat perbedaan yang signifikan secara statistik antara nilai *quantization error* dan nilai *topographic error* dari algoritme AMSOM dengan algoritme SOM. Metode AMSOM menghasilkan rata-rata nilai *quantization error* 27 kali lebih kecil dan rata-rata nilai *topographic error* 54 kali lebih kecil dibandingkan dengan metode SOM.

**Kata kunci:** pengelompokan, visualisasi, *self-organizing maps*, *adaptive moving self-organizing maps*,

## IMPROVED STUDENT GROUPING PERFORMANCE BASED ON LEARNING ACTIVITIES ON DIGITAL LEARNING MEDIA USING THE ADAPTIVE MOVING SELF-ORGANIZING MAPS METHOD

### Abstract

The digitization of the learning process makes it possible to produce recordings of each student's activity during learning. The resulting record can be used to group students based on the pattern of the learning process. The grouping results can be used to make adjustments to the learning components or learning methods for students. One of the most frequently used clustering methods is *Self-Organizing Maps* (SOM), SOM is a neural network method to maintain data topology when multidimensional input data is converted into output data with lower dimensions. The SOM neurons in the input dimension are updated throughout the training process, while the neurons in the output dimension do not get updated at all, this causes the neuron structure used in the initialization stage to remain the same until the end of the grouping process. In this study, the *Adaptive Moving Self-Organizing Maps* (AMSOM) method uses a more flexible neuron structure, allowing for the transfer, addition and deletion of neurons using 12 assignments of data from MONSAKUN learning media. The results showed that there was a statistically significant difference between the quantization error and the topographic error of the AMSOM algorithm and the SOM algorithm. The AMSOM method produces an average quantization error 27 times smaller and an average topographic error 54 times smaller than the SOM method.

**Keywords:** clustering, visualization, *self-organizing maps*, *adaptive moving self-organizing maps*

## 1. PENDAHULUAN

Perkembangan dunia pendidikan semakin hari semakin meningkat pesat, salah satu perkembangan yang terjadi adalah digitalisasi proses pembelajaran, digitalisasi proses pembelajaran akan menghasilkan data aktivitas dari kegiatan pembelajaran tersebut. Data yang dihasilkan tersebut dapat digunakan untuk mengungkap pola dari proses belajar yang dilakukan oleh siswa, pengenalan pola dari proses pembelajaran ini perlu dilakukan mengingat pentingnya penyesuaian komponen pembelajaran ataupun metode pembelajaran bagi siswa berdasarkan perilaku dan tingkat pengetahuan mereka (Ahmad, et al., 2015). Metode pengelompokan yang dapat digunakan untuk melakukan pengelompokan data aktivitas proses pembelajaran diperlukan, sehingga didapatkan pola-pola dari proses pembelajaran.

*Self-Organizing Maps* (SOM) merupakan salah satu jaringan syaraf tiruan dengan tujuan utama dari SOM ialah untuk mempertahankan topologi data ketika data *input* multidimensi diubah menjadi data *output* dengan dimensi yang lebih rendah (Hartono & Ogawa, 2017). SOM sangat baik dalam visualisasi data dengan struktur kompleks yang sulit dikenali manusia (Baez-Monroy & O'Keefe, 2005). Kelebihan dari metode SOM terletak pada implementasi yang sederhana, kecepatan eksekusi dan waktu pelatihan yang singkat, serta menggunakan basis *unsupervised learning* sehingga tidak dibutuhkan bimbingan untuk melakukan pelatihan data dan tidak diperlukan adanya target *output* (Cabada, et al., 2011) (Indriyani & Irfiani, 2019) (Tatoian & Hamel, 2018). Sebelumnya Pellicer-Chenoll (Pellicer-Chenoll, et al., 2015) telah melakukan penelitian untuk mengelompokkan siswa dengan menggunakan metode SOM. Pada penelitian ini, SOM digunakan untuk menginvestigasi keterkaitan antara aktivitas dan kebugaran fisik dengan performa akademik. Data aktivitas fisik dipilih karena aktivitas fisik sangat sering dikaitkan dengan kesehatan, dan kesehatan sangat penting bagi pelajar. Penelitian ini berhasil menemukan keterkaitan-keterkaitan antara aktivitas fisik dan performa akademik. Ahmad (Ahmad, et al., 2015) menggolongkan data aktivitas pelajar pada data *E-learning* menggunakan SOM. Hasil analisis menemukan pola dari tingkah laku pelajar sehingga dapat memberikan saran kepada pengajar tentang apa yang dibutuhkan oleh pelajar. Youngjin (Youngjin, 2018) menggunakan SOM dengan data log file *massive open online course* yang berisi berbagai data siswa dalam memecahkan tugas mingguan dan quiz. Penelitian mengidentifikasi kluster dari pelajar yang memiliki pola *problem-solving* yang sama.

Algoritme SOM yang digunakan pada penelitian tersebut merupakan algoritme SOM klasik yang memiliki struktur *neuron* yang tetap dan didefinisikan pada awal algoritme. Pada kasus dengan data set yang karakteristiknya tidak diketahui dengan baik akan sulit untuk menentukan struktur yang tepat agar hasil yang diinginkan dicapai (Spanakis & Weiss,

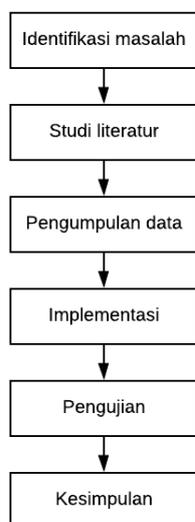
2016). Oleh karena itu, pada penelitian ini dilakukan percobaan melakukan klusterisasi menggunakan varian algoritme SOM dengan nama *Adaptive Moving Self-Organizing Maps* (AMSOM). *Neuron* pada AMSOM dimungkinkan untuk berubah posisi selama proses pelatihan sehingga akan memberikan visualisasi yang lebih baik, serta waktu pelatihan yang lebih cepat. Varian algoritme ini juga memungkinkan dilakukannya penambahan dan penghapusan *neuron* (menyesuaikan terhadap data set) sehingga diharapkan dapat merepresentasikan data set dengan lebih efisien. Algoritme AMSOM telah digunakan oleh Suwalka (Suwalka & Agrawal, 2018) (Suwalka & Agrawal, 2019) dalam pencitraan medis untuk melakukan deteksi penyakit *neurodegenerative* dan identifikasi tumor. Hasil dari penelitian menunjukkan AMSOM memberikan hasil yang cukup baik dan dapat mencapai target yang diharapkan.

Berdasarkan latar belakang tersebut, maka pada penelitian ini dilakukan pengelompokan siswa berdasarkan aktivitas belajar yang dilakukan pada media pembelajaran interaktif MONSAKUN menggunakan algoritme AMSOM. Tujuan dari penelitian ini adalah menentukan nilai parameter-parameter pada Algoritme AMSOM yang dapat memberikan hasil paling optimal terhadap pengelompokan pelajar berdasarkan pola berpikirnya. Selain itu, penelitian ini bertujuan untuk membandingkan hasil pengelompokan pelajar menggunakan algoritme *Adaptive Moving Self-Organizing Maps* dengan algoritme *Self-Organizing Maps* menggunakan metode evaluasi *Quantization Error* (QE) dan *Topographic Error* (TE).

## 2. METODE PENELITIAN

Tahapan pertama yang dilakukan dalam penelitian ini adalah melakukan identifikasi masalah, dimana melalui studi literatur ditemukan bahwa perlunya dilakukan penyesuaian komponen pembelajaran ataupun metode pembelajaran bagi siswa berdasarkan perilaku dan tingkat pengetahuan mereka (Ahmad, et al., 2015). Berdasarkan studi literatur tersebut maka pengelompokan siswa dilakukan berdasarkan aktivitasnya pada saat proses pembelajaran. Data aktivitas yang digunakan pada penelitian ini adalah data aktivitas siswa dalam proses belajar pada media pembelajaran MONSAKUN yang tersimpan dalam bentuk file datalog. Tahapan selanjutnya adalah melakukan perancangan dari penerapan algoritme. Setelah selesai melakukan desain terhadap rancangan maka selanjutnya dilakukan pengimplementasian hasil rancangan tersebut ke dalam *script* pemrograman. Tahapan selanjutnya adalah melakukan pengujian terhadap parameter-parameter yang digunakan pada algoritme dan perbandingan hasil penerapan algoritme AMSOM dengan hasil penerapan algoritme SOM, pengujian dan perbandingan dilakukan menggunakan dua metode evaluasi yakni, *Quantization Error* dan

*topographic error*. Tahapan-tahapan yang dilakukan pada penelitian ini gambarkan pada Gambar 1.



Gambar 1. Diagram alir penelitian

## 2.1. MONSAKUN Sebagai Media Pembelajaran Digital

Monsakun merupakan media pembelajaran digital interaktif dengan pendekatan *problem-posing* (merumuskan masalah atau membuat masalah) dalam permasalahan kata aritmatika sebagai integrasi kalimat (Hasanah, et al., 2015) (Hirashima, et al., 2014). Dalam kegiatan *problem-posing*, peserta didik tidak membuat rumusan masalah mereka sendiri, namun mereka dituntut untuk menginterpretasikan kartu kalimat dan mengintegrasikannya ke dalam sebuah masalah. Proses tersebut dikenal dengan istilah “*problem-posing as sentence-integration*” (Hirashima, et al., 2007).

Media pembelajaran MONSAKUN digunakan pada kelas dengan total 39 siswa kelas satu dari sekolah dasar Hiroshima Fuzoku (Hasanah, et al., 2015). Permasalahan dalam media pembelajaran MONSAKUN dibagi menjadi 5 level, level 1, 3 dan 5 digunakan dalam 2 sesi kelas, sedangkan level 2 dan 4 hanya digunakan pada satu sesi. Level 1, 3 dan 5 memiliki 12 *assignments* dengan 4 tipe cerita: kombinasi (1-3), penambahan (4-6), pengurangan (7-9) dan perbandingan (10-12) (Supianto, et al., 2017). Pada level 1 siswa diberikan rumus permasalahan dan diminta untuk mengajukan permasalahan *forward thinking*. Pada level 3 siswa diberikan rumus permasalahan dan diminta untuk mengajukan permasalahan *reverse thinking*. Pada level 5, siswa diberikan rumus perhitungan dan diminta untuk mengajukan permasalahan *reverse thinking*, level 5 merupakan level dengan *assignment* paling menantang pada MONSAKUN (Hasanah, et al., 2015). Pada penelitian ini digunakan hanya data log aktivitas level 5 dari sistem pembelajaran MONSAKUN. Hal ini dikarenakan rata-rata langkah dan kesalahan pada level 5 jauh lebih tinggi jika dibandingkan dengan level 1 hingga level 4, sehingga

dapat dikatakan bahwa level 5 lebih menantang jika dibandingkan level lainnya (Supianto, et al., 2017).

## 2.2. Adaptive Moving Self-Organizing Maps

*Adaptive Moving Self-Organizing Maps* (AMSOM), merupakan metode yang dikembangkan dari metode *Self-Organizing Maps* (SOM). Perbedaan utama antara algoritme AMSOM dan SOM terletak pada fleksibilitas struktur *neuron*. Pada AMSOM, pada akhir setiap epoch setelah dilakukan pembaharuan terhadap nilai vektor bobot, dilakukan pembaharuan terhadap posisi *neuron* sehingga dapat memberikan visualisasi yang lebih baik serta mempersingkat waktu pelatihan. Pada AMSOM juga dimungkinkan dilakukan penambahan dan penghapusan *neuron*, proses penambahan dan penghapusan *neuron* ini didasarkan pada kebutuhan dataset. Secara umum algoritme AMSOM adalah sebagai berikut:

**Langkah 1:** Penentuan nilai parameter AMSOM yang terdiri dari ukuran *grid*, *sigma* ( $\sigma$ ), *gamma* ( $\gamma$ ), *alpha* ( $\alpha$ ), *agemax*, *tadd*, *spread factor* dan jumlah *epoch*.

**Langkah 2:** Inisialisasi nilai vektor bobot ( $w$ ) dengan nilai acak (dengan ukuran  $M \times D$ ), inisialisasi nilai vektor posisi ( $r$ ) dengan nilai koordinat dari *neuron* pada dimensi output berdasarkan struktur *grid* yang digunakan (dengan ukuran  $M \times O$ ), inisialisasi matrik *edge connectivity* ( $E$ ) dan matrik *edge age* ( $A$ ), dimana matrik  $E$  dan  $A$  merupakan matrik ortogonal dan simetris (dengan ukuran  $M \times M$ ).  $E(p, q)$  akan bernilai 1 jika *neuron*  $p$  dan  $q$  terhubung dan 0 jika tidak,  $A(p, q)$  menunjukkan total *epoch* *neuron*  $p$  dan  $q$  tetap terhubung.  $M$  adalah jumlah *neuron*,  $D$  merupakan jumlah dimensi data masukan dan  $O$  adalah jumlah dimensi data keluaran.

**Langkah 3:** Inisialisasi nilai *growing threshold* menggunakan persamaan (1), pada persamaan,  $GT$  merupakan *growing threshold*.  $D$  merupakan jumlah dimensi dari data masukan dan  $SF$  merupakan nilai dari parameter *spread factor*.

**Langkah 4:** Pencarian *best matching neuron* 1 dan *best matching neuron* 2 menggunakan persamaan (2), pada persamaan,  $c$  merupakan *best matching neuron* dan  $d_i(t)$  dihitung menggunakan persamaan (3), pada persamaan,  $d_i$  merupakan *distance* antara *neuron*  $i$  dan data  $\bar{x}$ .  $\bar{x}$  merupakan data masukan dan  $\bar{w}_i$  merupakan vektor bobot dari *neuron*  $i$ .

**Langkah 5:** Pembaharuan nilai matrik  $E$  dan  $A$ .

**Langkah 6:** Pembaharuan nilai vektor bobot ( $\bar{w}$ ) menggunakan persamaan (4), pada persamaan,  $\bar{w}_i(t+1)$  merupakan nilai vektor bobot *neuron*  $i$  setelah di perbaharui.  $n_j(t)$  adalah jumlah data yang memiliki *neuron*  $j$  sebagai pemenangnya,  $h_{ji}(t)$  dihitung menggunakan persamaan (5) dan  $\bar{x}_j(t)$  merupakan nilai rata-rata dari data yang memiliki *neuron*  $j$  sebagai pemenangnya.  $h_{ji}(t)$  merupakan fungsi ketetanggaan.  $\bar{r}_j$  dan  $\bar{r}_i$  adalah koordinat posisi

dari node  $j$  dan  $i$ ,  $\sigma(t)$  adalah radius dari ketetangaan.

**Langkah 7:** Pembaharuan nilai vektor posisi ( $r$ ) menggunakan persamaan (6), pada persamaan,  $\bar{r}_i(t+1)$  merupakan nilai posisi bobot *neuron*  $i$  setelah di perbaharui.  $r_i$  dan  $r_j$  adalah koordinat posisi dari *neuron*  $i$  dan *neuron*  $j$ ,  $\alpha(t)$  adalah *learning rate*,  $\delta_{ji}(t)$  dihitung menggunakan persamaan (7), pada persamaan,  $\delta_{ji}(t)$  merupakan fungsi ketetangaan.  $\bar{w}_j$  dan  $\bar{w}_i$  adalah vektor bobot dari *neuron*  $j$  dan  $i$ ,  $\gamma$  adalah parameter yang mengontrol penyusutan ketetangaan dan  $\sigma(t)$  adalah radius dari ketetangaan.

**Langkah 8:** Melakukan penambahan dan penghapusan *neuron*. *neuron* baru hanya akan ditambahkan jika *neuron* dengan nilai *quantization error* paling tinggi telah lebih besar dari nilai GT. Nilai *quantization error* dihitung menggunakan persamaan (8), pada persamaan,  $E_c$  merupakan nilai *quantization error*,  $\bar{x}_i$  merupakan data ke  $i$  dan  $\bar{w}_c$  *neuron* pemenang dari data ke  $i$ . Penghapusan *neuron* akan dilakukan jika suatu *neuron* tidak memiliki hubungan dengan *neuron* lainnya.

**Langkah 9:** Melakukan proses pengelompokan *neuron* dimana jika nilai  $dist(i,j) \leq v$  maka hubungan *neuron*  $i$  dan  $j$  bernilai 1, jika  $dist(i,j) > v$  maka hubungan *neuron*  $i$  dan  $j$  bernilai 0. *neuron* yang saling terhubung dikelompokkan menjadi satu kelompok. Nilai  $v$  dihitung menggunakan persamaan (9), pada persamaan,  $v$  merupakan *threshold*.  $N$  merupakan jumlah *neuron* dan  $dist(i,j)$  merupakan jarak *neuron*  $i$  dan  $j$  pada dimensi data masukan.

**Langkah 10:** Melakukan pengukuran kualitas map menggunakan metode *Quantization Error* (QE) (mengukur seberapa bagus kecocokan antara map dan data masukan) menggunakan persamaan (10), dimana  $N$  adalah jumlah data masukan dan  $\|\bar{x}_i - \bar{w}_c\|$  adalah *euclidean distance* dari vektor masukan ke *neuron* pemenang dari vektor masukan. Metode kedua yang digunakan untuk mengukur kualitas dari map adalah *Topographic error* (TE) (mengukur seberapa baik topologi tetap dipertahankan oleh map) menggunakan persamaan (11), dimana  $u(x_k)$  akan bernilai 1 jika BMN 1 dan BMN 2 dari data  $\bar{x}$  terkoneksi, jika tidak  $u(x_k)$  akan bernilai 0. AMSOM dengan nilai QE dan TE yang lebih kecil memiliki tingkat keakuratan yang lebih tinggi dibandingkan AMSOM dengan nilai QE dan TE yang lebih besar (Gabrielsson & gabrielsson, 2006).

$$GT = -\ln(D) \times \ln(SF) \quad (1)$$

$$c = \{i, \min_i d_i(t)\} \quad (2)$$

$$d_i(t) = \|\bar{x}(t) - \bar{w}_i\|^2 \quad (3)$$

$$\bar{w}_i(t+1) = \frac{\sum_j n_j(t) \cdot h_{ji}(t) \cdot \bar{x}_j(t)}{\sum_j n_j(t) \cdot h_{ji}(t)} \quad (4)$$

$$h_{ji}(t) = \exp\left(-\frac{\|\bar{r}_j - \bar{r}_i\|^2}{\sigma(t)^2}\right) \quad (5)$$

$$\bar{r}_i(t+1) = \bar{r}_i(t) + \alpha(t) \cdot \frac{\sum_j n_j(t) \cdot \delta_{ji}(t) \cdot (\bar{r}_j(t) - \bar{r}_i(t))}{\sum_j n_j(t) \cdot \delta_{ji}(t)} \quad (6)$$

$$\delta_{ji}(t) = \exp\left(-\frac{\|\bar{w}_j - \bar{w}_i\|^2}{\gamma \times \sigma(t)^2}\right) \quad (7)$$

$$E_c = \|\bar{x}_i - \bar{w}_c\| \quad (8)$$

$$v = \frac{1}{N} \sum_{i=1}^N dist(i, j) \quad (9)$$

$$E_q = \frac{1}{N} \sum_{i=1}^N \|\bar{x}_i - \bar{w}_c\| \quad (10)$$

$$E_t = \frac{1}{N} \sum_{k=1}^N Nu(x_k) \quad (11)$$

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Hasil Pengujian Parameter

Pada bagian ini, dilakukan pengujian terhadap parameter ukuran *grid*, *gamma*, *alpha*, *agemax*, *tadd*, *spread factor* dan jumlah *epoch* yang digunakan pada metode AMSOM, yang bertujuan untuk mendapatkan nilai parameter optimal pada metode AMSOM. pengujian ini bertujuan untuk menjawab tujuan penelitian pertama mengenai penentuan nilai parameter-parameter pada Algoritme AMSOM yang dapat memberikan hasil paling optimal terhadap pengelompokan pelajar berdasarkan pola berpikirnya.

##### 3.1.1 Hasil Pengujian Parameter Ukuran Grid dan Sigma

Pengujian parameter ukuran *grid* dan *sigma* digunakan untuk mencari nilai parameter ukuran *grid* dan *sigma* optimal, yang dapat memberikan nilai QE terkecil pada proses pengelompokan siswa. Pengujian parameter ukuran *grid* dimulai dari ukuran 4x4 hingga ukuran 7x7, untuk setiap ukuran *grid* dilakukan pengujian parameter *sigma* dengan rentang nilai 1 hingga 7. Proses pengujian dilakukan sebanyak 10 kali untuk setiap kombinasi nilai parameter ukuran *grid* dan parameter *sigma*. Parameter ukuran *grid* dan *sigma* dengan rata-rata nilai QE terendah dianggap sebagai nilai parameter optimal. Hasil proses pengujian parameter ukuran *grid* dan *sigma* menggunakan data *assignment* 1 – *assignment* 12 ditampilkan pada Tabel 1.

Tabel 1. Hasil Pengujian Parameter *Grid Size* dan *Sigma*

<i>assignment</i>	<i>grid size</i>	<i>sigma</i>
1	7x7	2
2	7x7	2
3	7x7	2
4	7x7	2
5	7x7	2
6	7x7	2
7	7x7	2
8	7x7	2
9	7x7	2
10	7x7	2
11	7x7	2
12	7x7	2

Berdasarkan Tabel 1, parameter *grid size* optimal untuk semua *assignment* adalah 7x7. Nilai ini diperoleh karena ukuran *grid* yang semakin besar akan membuat persebaran *neuron* awal menjadi semakin beragam, sehingga bentuk map akan dengan cepat menyesuaikan struktur data. Penyesuaian tersebut menyebabkan semakin kecilnya nilai *quantization error*, karena nilai *quantization error* menunjukkan seberapa dekat data dengan *best matching neuron*. Untuk parameter nilai *sigma* nilai optimal yang didapat adalah 2 untuk semua *assignment*. Nilai ini diperoleh dikarenakan keterkaitan antara parameter *sigma* dan parameter ukuran *grid*. *Sigma* dengan nilai 2 menyebabkan hanya tetangga terdekat dari *neuron* pemenang yang akan di perbaharui pada beberapa *epoch* awal, dan akan semakin kecil setiap peningkatan *epoch*. Nilai *sigma* ini memberikan kombinasi optimal dengan ukuran *grid* 7x7.

### 3.1.2 Hasil Pengujian Parameter Gamma

Pengujian parameter *gamma* digunakan untuk mencari nilai parameter *gamma* optimal, yang dapat memberikan nilai *quantization error* terkecil pada proses pengelompokan siswa. Pengujian parameter *gamma* menggunakan nilai 1 sampai 10, berkelipatan 1. Proses pengujian dilakukan sebanyak 10 kali untuk setiap nilai parameter *gamma*, perulangan ini dilakukan dikarenakan terdapatnya penggunaan nilai acak untuk inisialisasi vektor bobot pada algoritme. Parameter *gamma* yang menghasilkan rata-rata nilai *quantization error* yang terendah dianggap sebagai nilai parameter optimal. Hasil proses pengujian parameter *gamma* menggunakan data *assignment 1 – assignment 12* ditampilkan pada Tabel 2.

Parameter *gamma*, merupakan parameter yang mempengaruhi penyusutan ketetanggaan. Sehingga nilai optimal parameter *gamma* akan menyesuaikan terhadap struktur data dan struktur *neuron*. Hal tersebut menyebabkan terjadinya perbedaan nilai *sigma* optimal yang diperoleh untuk tiap-tiap *assignment*.

Tabel 2. Hasil Pengujian Parameter Gamma

<i>assignment</i>	<i>gamma</i>
1	1
2	6
3	6
4	1
5	6

6	5
7	4
8	1
9	8
10	3
11	3
12	3

### 3.1.3 Hasil Pengujian Parameter Alpha

Pengujian parameter *alpha* digunakan untuk mencari nilai parameter *alpha* optimal, yang dapat memberikan nilai *quantization error* terkecil pada proses pengelompokan siswa. Pengujian parameter *alpha* menggunakan nilai 0,001 sampai 1. Proses pengujian dilakukan sebanyak 10 kali untuk setiap nilai parameter *alpha*. Perulangan ini dilakukan dikarenakan terdapatnya penggunaan nilai acak untuk inisialisasi vektor bobot pada algoritme. Nilai parameter *alpha* yang menghasilkan rata-rata nilai *quantization error* yang terendah dianggap sebagai nilai parameter optimal. Hasil proses pengujian parameter *alpha* menggunakan data *assignment 1 – assignment 12* ditampilkan pada Tabel 3.

Pada parameter *alpha*, nilai optimal yang didapatkan untuk semua *assignment* tidak ada yang melebihi dari 0,01, dengan nilai dominan 0,004 dan 0,007. Hal ini disebabkan karena nilai *alpha* yang kecil mengakibatkan perpindahan *neuron* pada ruang *output* memiliki nilai yang juga kecil. Nilai yang kecil ini sesuai dengan struktur inisialisasi awal neuron pada ruang *output* yang memiliki jarak 1.

Tabel 3. hasil Pengujian Parameter Alpha

<i>assignment</i>	<i>alpha</i>
1	0,004
2	0,010
3	0,001
4	0,004
5	0,004
6	0,004
7	0,001
8	0,004
9	0,007
10	0,007
11	0,004
12	0,007

### 3.1.4 Hasil Pengujian Parameter Agemax

Pengujian parameter *agemax* digunakan untuk mencari nilai parameter *agemax* optimal, yang dapat memberikan nilai *quantization error* terkecil pada proses pengelompokan siswa. Pengujian parameter *agemax* menggunakan nilai 10 sampai 100, dengan kelipatan 10. Proses pengujian dilakukan sebanyak 10 kali untuk setiap nilai parameter *agemax*, perulangan ini dilakukan dikarenakan terdapatnya penggunaan nilai acak untuk inisialisasi vektor bobot pada algoritme. Parameter *agemax* yang menghasilkan rata-rata nilai *quantization error* yang terendah dianggap sebagai nilai parameter optimal. Hasil proses pengujian parameter *agemax* menggunakan data *assignment 1 – assignment 12* ditampilkan pada Tabel 4.

Tabel 4. Hasil Pengujian Parameter Agemax

assignment	Age <sub>max</sub>
1	80
2	30
3	90
4	50
5	100
6	90
7	90
8	70
9	50
10	70
11	30
12	80

Pada parameter *agemax*, nilai optimal yang diperoleh berkisar antara *interval* 30 sampai dengan 100 dengan jumlah yang hampir merata. Nilai ini diperoleh karena nilai tersebut telah memberikan cukup waktu untuk struktur *grid* menyesuaikan terhadap struktur data.

### 3.1.5 Hasil Pengujian Parameter Tadd

Pengujian parameter *tadd* digunakan untuk mencari nilai parameter *tadd* optimal, yang dapat memberikan nilai *quantization error* terkecil. Pengujian parameter menggunakan nilai 10 sampai 100, dengan kelipatan 10. Proses pengujian dilakukan sebanyak 10 kali untuk setiap nilai parameter *tadd*, perulangan ini dilakukan dikarenakan terdapatnya penggunaan nilai acak untuk inisialisasi vektor bobot pada algoritme. Parameter *tadd* yang menghasilkan rata-rata nilai *qe* yang terendah dianggap sebagai nilai parameter optimal. Hasil proses pengujian parameter *tadd* menggunakan data *assignment 1 – assignment 12* ditampilkan pada Tabel 5.

Tabel 5. Hasil Pengujian Parameter Tadd

assignment	t <sub>add</sub>
1	10
2	30
3	50
4	30
5	80
6	10
7	60
8	30
9	30
10	30
11	80
12	30

Parameter *tadd* untuk semua *assignment* memberikan nilai yang bervariasi antara 10, 30, 50, 60 dan 80, dengan nilai 30 sebagai nilai dominan. Pada parameter *tadd*, pada beberapa *assignment*, *quantization error* yang didapatkan dari nilai *tadd* yang berbeda menunjukkan hasil yang sama, pada kasus tersebut parameter *tadd* yang mendekati nilai 30 akan dipilih sebagai nilai optimal. Pemilihan tersebut dikarenakan 30 epoch diharapkan dapat memberikan hasil optimal (Spanakis & Weiss, 2017).

### 3.1.6 Hasil Pengujian Parameter Spread Factor

Pengujian parameter *spread factor* digunakan untuk mencari nilai parameter *spread factor* optimal,

yang dapat memberikan nilai *quantization error* terkecil pada proses pengelompokan siswa. Pengujian parameter *spread factor* menggunakan nilai 0,1 sampai 0,7, dengan kelipatan 0,1. Proses pengujian dilakukan sebanyak 10 kali untuk setiap nilai parameter *spread factor*, perulangan ini dilakukan dikarenakan terdapatnya penggunaan nilai acak untuk inisialisasi vektor bobot pada algoritme. Parameter *spread factor* yang menghasilkan rata-rata nilai *quantization error* yang terendah dianggap sebagai nilai parameter optimal. Hasil proses pengujian parameter *spread factor* menggunakan data *assignment 1 – assignment 12* ditampilkan pada Tabel 6.

Pada parameter *spread factor*, beberapa *assignment*, *quantization error* yang didapatkan dari nilai *spread factor* yang berbeda menunjukkan hasil yang sama, pada kasus tersebut, nilai parameter *spread factor* yang mendekati nilai 0,5 akan dipilih sebagai nilai optimal. Pemilihan tersebut dikarenakan *spread factor* dengan nilai 0,5 selalu menghasilkan hasil yang memuaskan (Spanakis & Weiss, 2017).

Tabel 6. Hasil Pengujian Parameter Spread Factor

assignment	spread factor
1	0,5
2	0,5
3	0,5
4	0,5
5	0,5
6	0,5
7	0,5
8	0,5
9	0,5
10	0,5
11	0,5
12	0,5

### 3.1.7 Hasil Pengujian Parameter Epoch

Pengujian parameter *epoch* digunakan untuk mencari nilai parameter *epoch* optimal, yang dapat memberikan nilai *quantization error* terkecil pada proses pengelompokan siswa. Pengujian parameter *epoch* menggunakan nilai 100 sampai 1000, dengan kelipatan 100. Proses pengujian dilakukan sebanyak 10 kali untuk setiap nilai parameter *epoch*, perulangan ini dilakukan dikarenakan terdapatnya penggunaan nilai acak untuk inisialisasi vektor bobot pada algoritme. Parameter *epoch* yang menghasilkan rata-rata nilai *quantization error*. Hasil proses pengujian parameter *epoch* menggunakan data *assignment 1 – assignment 12* ditampilkan pada Tabel 7.

Tabel 7. Hasil Pengujian Parameter Epoch

assignment	epoch
1	500
2	500
3	1000
4	500
5	500
6	400
7	500
8	300
9	400
10	500

assignment	epoch
11	500
12	500

Pada parameter *epoch*, nilai optimal yang diperoleh bervariasi antara 300, 400, 500 dan 1000, dengan 8 dari 10 *assignment* memiliki nilai *epoch* 500 sebagai nilai optimal. Dominasi ini disebabkan karena parameter *epoch* merupakan parameter terakhir yang diuji, sehingga nilai parameter 500 sebagai nilai inialisasi membuat parameter lainnya optimal pada saat nilai parameter *epoch* 500.

### 3.2. Hasil Pengujian Perbandingan Pengelompokan Menggunakan Algoritme AMSOM dengan Algoritme SOM

Pada pengujian ini dilakukan perbandingan antara hasil pengelompokan menggunakan algoritme *error* dan jumlah *neuron*. Nilai variabel yang diuji merupakan nilai rata-rata dari 10 kali percobaan, penggunaan nilai rata-rata ini diharapkan dapat mengurangi pengaruh yang diberikan oleh inialisasi nilai acak pada vektor bobot. Hasil perbandingan pengelompokan algoritme AMSOM dengan

pengelompokan menggunakan algoritme SOM ditunjukkan pada Tabel 8

Nilai QE, TE dan jumlah *neuron* yang telah didapatkan dari hasil pengujian pengelompokan menggunakan algoritme *Adaptive Moving Self-Organizing Maps* dengan pengelompokan menggunakan algoritme *Self-Organizing Maps* kemudian selanjutnya melalui pengujian normalitas menggunakan metode *Shapiro-Wilks* untuk menentukan metode perbandingan yang akan digunakan. Hasil tes normalitas terhadap nilai QE, TE dan jumlah *neuron* ditunjukkan pada Tabel 9.

Tabel 8. Hasil Pengujian Normalitas

	AMSOM	SOM
<i>Quantization Error</i>	0,05257	0,51683
<i>Topographic Error</i>	0,00001	0,86157
Jumlah <i>Neuron</i>	0,02764	0,00000

Tabel 9. hasil pengujian perbandingan metode AMSOM dan metode SOM

	statistic (U)	p-value
<i>Quantization Error</i>	0.000000	0.000018
<i>Topographic Error</i>	0.000000	0.000012
Jumlah <i>Neuron</i>	0.000000	0.000005

Tabel 10. Perbandingan Hasil Pengelompokan Menggunakan Algoritme AMSOM Dengan Pengelompokan Menggunakan Algoritme SOM

assignment	karakteristik		QE		TE		# neuron	
	instances	fitur	AMSOM	SOM	AMSOM	SOM	AMSOM	SOM
1	39	12	0,01691	0,23526	0,01538	0,21795	47,9	49
2	38	12	0,00301	0,05904	0,02105	0,08684	44,2	49
3	39	12	0,00092	0,14078	0,00000	0,10513	42,8	49
4	39	12	0,01307	0,17396	0,00000	0,20769	47,3	49
5	39	12	0,00385	0,09098	0,00000	0,23846	48	49
6	37	12	0,01024	0,16081	0,00000	0,17027	47,4	49
7	35	12	0,00755	0,24823	0,00000	0,32571	48,3	49
8	32	12	0,00214	0,12183	0,00000	0,19375	47,7	49
9	30	12	0,00289	0,11604	0,00000	0,10000	46,2	49
10	28	12	0,00472	0,12148	0,00000	0,11429	46,1	49
11	24	12	0,00027	0,23526	0,00000	0,21795	42,8	49
12	24	12	0,00298	0,16362	0,00000	0,02083	46,7	49

AMSOM dengan pengelompokan menggunakan algoritme SOM. Perbandingan dilakukan untuk membuktikan bahwa pengelompokan menggunakan algoritme *Adaptive Moving Self-Organizing Maps* dapat memberikan hasil yang lebih baik dibandingkan pengelompokan menggunakan algoritme *Self-Organizing Maps*. Pengujian juga bertujuan untuk menjawab tujuan penelitian kedua, membandingkan hasil pengelompokan pelajar menggunakan algoritme *Adaptive Moving Self-Organizing Maps* dengan algoritme *Self-Organizing Maps* menggunakan metode evaluasi *Quantization Error* (QE) dan *Topographic Error* (TE).

Variabel yang digunakan dalam pengujian antara lain adalah nilai *quantization error*, nilai *topographic*

Berdasarkan hasil pengujian normalitas data, ditemukan bahwa 3 dari 6 data memiliki nilai *p-value* yang lebih kecil dari 0.05 sehingga data dikategorikan tidak terdistribusi secara normal. Hasil tersebut menjadikan metode *Mann Whitney U Test* yang merupakan metode perbandingan *non-parametric* dipilih sebagai metode untuk melakukan

perbandingan hasil antara metode AMSOM dan SOM. Pemilihan tipe tes *non-parametric* juga didasari bahwa jika terdapat penyimpangan pada normalitas data, akan lebih baik jika menggunakan tipe tes *non-parametric* (Kitchen, 2009).

Hasil pengujian perbandingan pengelompokan menggunakan metode AMSOM dengan menggunakan metode SOM ditampilkan pada Tabel 10. Perbandingan ini menggunakan metode *Mann Whitney U Test* dengan nilai  $\alpha = 0.05$  dan nilai *critical value* = 37 (nilai *critical value* untuk *Mann Whitney U Test* dengan nilai  $n_1 = 12$ ,  $n_2 = 12$  dan *non directional*  $\alpha = 0.05$ ). Berdasarkan hasil perbandingan pada Tabel 10, dapat disimpulkan bahwa terdapat perbedaan yang signifikan secara statistik untuk variabel *Quantization Error*, variabel *Topographic Error* dan variabel jumlah *neuron* yang dihasilkan dari pengelompokan menggunakan algoritme AMSOM dan pengelompokan menggunakan algoritme SOM. Hal ini disebabkan nilai U pada variabel QE, TE dan jumlah *neuron*

memiliki nilai yang lebih kecil atau sama dengan nilai *critical value* (Nachar, 2008).

#### 4. KESIMPULAN

Implementasi algoritme AMSOM pada pengelompokan siswa menggunakan data aktivitas belajar pada 12 data *assignments* (dengan karakteristik yang berbeda pada jumlah siswa dan struktur dari permasalahan tiap *assignment*) menunjukkan bahwa terdapat perbedaan pada parameter optimal yang dihasilkan untuk tiap-tiap *assignment*, kecuali pada parameter ukuran *grid* dan *sigma* dan *spread factor*. Ketiga parameter menghasilkan nilai parameter optimal yang sama untuk tiap-tiap *assignment*. Algoritme AMSOM disimpulkan mengalami peningkatan pada performa jika dibandingkan dengan algoritme SOM biasa, dimana algoritme AMSOM menghasilkan *reference vector* yang lebih baik. Kesimpulan tersebut ditunjukkan oleh nilai QE dan TE algoritme AMSOM yang lebih kecil jika dibandingkan dengan algoritme SOM biasa.

#### DAFTAR PUSTAKA

- AHMAD, N. B., ISHAK, M. K., ALIAS, U. F. & MOHAMAD, N., 2015. An Approach for E-Learning Data Analytics using SOM Clustering. *Int. J. Advance Soft Compu*, 7(3), pp. 94-112.
- BAEZ-MONROY, V. O. & O'KEEFE, S., 2005. *Modelling Incremental Learning With The Batch SOM Training Method*. Rio de Janeiro, IEEE.
- CABADA, R. Z., ESTRADA, M. L. B. & GARCIA, C. A. R., 2011. EDUCA: A Web 2.0 Authoring Tool For Developing Adaptive and Intelligent Tutoring Systems Using a Kohonen Network. *Expert Systems with Applications*.
- GABRIELSSON, S. & GABRIELSSON, S., 2006. The use of Self-Organizing Maps in Recommender Systems.
- HARTONO, P. & OGAWA, K., 2017. Intuitive Analysis by Visualizing Context Relevant E-learning Data. *IPSJ Transactions on Computers and Education*, 3(2), pp. 20-27.
- HASANAH, N., HIRASHIMA, T. & HAYASHI, Y., 2015. *Investigation of Students' Performance in Monsakun Problem Posing Activity based on the Triplet Structure Model of Arithmetical Word Problems*. Hanzhou, s.n., pp. 27-36.
- HIRASHIMA, T., YAMAMOTO, S. & HAYASHI, Y., 2014. Triplet Structure Model of Arithmetical Word Problems. *International Conference on Human Interface and the Management of Information*, pp. 42-50.
- HIRASHIMA, T., YOKOYAMA, T. & OKAMOTO, M., 2007. Learning by Problem-Posing as Sentence-Integration and Experimental Use. *AIED*, pp. 254-261.
- INDRIYANI, F. & IRFIANI, E., 2019. Clustering Data Penjualan pada Toko Perlengkapan Outdoor Menggunakan Metode K-Means. *JUITA: Jurnal Informatika*, 7(2), pp. 109-113.
- KITCHEN, C. M., 2009. Nonparametric vs Parametric Tests of Location in Biomedical Research. *American Journal of Ophthalmology*, 147(4), pp. 571-572.
- NACHAR, N., 2008. The Mann-Whitney U: A Test for Assessing Whether Two Independent Samples Come. *Tutorials in Quantitative Methods for Psychology*, 4(1), pp. 13-20.
- PELLICER-CHENOLL, M., GARCIA-MASSO, X., MORALES, J., SERRA-ANO, P., SOLANA-TRAMUNT, M., GONZALEZ, L. M., TOCA-HERRERA, J. L., 2015. Physical activity, physical fitness and academic achievement in adolescents: a self-organizing maps approach. *Health Education Research*, 30(3), pp. 436-448.
- SPANAKIS, G. & WEISS, G., 2016. AMSOM: Adaptive Moving Self-organizing Map for Clustering and Visualization. *8th International Conference on Agents and Artificial Intelligence*.
- SPANAKIS, G. & WEISS, G., 2017. Enhancing Visual Clustering Using Adaptive Moving Self-Organizing Maps (AMSOM). *van den Herik J., Filipe J. (eds) Agents and Artificial Intelligence. ICAART 2016. Lecture Notes in Computer Science*, Volume 10162, pp. 189-211.
- SUPIANTO, A. A., HAYASHI, Y. & HIRASHIMA, T., 2017. An Investigation of Learner's Actions in Posing Arithmetic Word. *IEICE Transactions on Information and Systems*, 100(11), pp. 2725-2728.
- SUPIANTO, A. A., HAYASHI, Y. & HIRASHIMA, T., 2017. Process-based Assignment-Setting Change for Support of Overcoming Bottlenecks in Learning by Problem-Posing in Arithmetic Word Problems. *Journal of Physics: Conference Series*, 812(1).
- SUWALKA, I. & AGRAWAL, N., 2018. An improved unsupervised mapping technique using AMSOM for neurodegenerative disease detection. *Int. J. Computational Systems Engineering*, 4(2/3), pp. 185-194.
- SUWALKA, I. & AGRAWAL, N., 2019. Machine learning using AMKSOM for. *2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP)*, pp. 1-7.
- TATOIAN, R. & HAMEL, L., 2018. Self-Organizing Map Convergence. *International Journal of Service Science, Management, Engineering, and Technology (IJSSMET)*, 9(2), pp. 61-84.

YOUNGJIN, L., 2018. Using Self-Organizing Map and Clustering to Investigate Problem-Solving Patterns in the Massive Open Online Course: An Exploratory Study. *Journal of Educational Computing Research* , 0(0), pp. 1-20.

*Halaman ini sengaja dikosongkan*