KONSTRUKSI PERSEGI AJAIB DENGAN ENTRI BILANGAN BULAT

(*MAGIC SQUARE CONSTRUCTION WITH INTEGER ENTRY*)

Ulil Albab Mu'min[1], Bib Paruhum Silalahi[2]
[1]Institut Pertanian Bogor, ulilalbabmumin@apps.ipb.ac.id
[2] Institut Pertanian Bogor, bibsi@apps.ipb.ac.id

Abstrak

Persegi ajaib adalah kotak-kotak persegi berisi bilangan berbeda yang disusun sedemikian rupa sehingga jumlah bilangan-bilangan pada baris, kolom, dan diagonal adalah sama. Penelitian ini membahas tentang pola dan algoritma untuk menyusun persegi ajaib berukuran $m \times m$ dari rangkaian m^2 bilangan bulat berurutan. Konstruksi algoritma dibagi menjadi tiga kasus, yaitu: algoritma persegi ajaib ordo ganjil $(2j + 1) \times (2j + 1)$, algoritma persegi ajaib ordo genap $(4j) \times (4j)$, dan algoritma persegi ajaib ordo genap $(4j + 2) \times (4j + 2)$ dengan $j = 1,2, \dots, m$.

Kata kunci: Algoritma, Bilangan Bulat, Persegi Ajaib

Abstract

Magic square is a square of different numbers arranged so that the sum of the numbers in the row, column, and diagonal is the same. This research discusses the pattern and algorithm for constructing the magic square order $m \times m$ *from a series of* m^2 *consecutive integers. The algorithm construction is divided into three cases, namely: algorithm odd order magic square* $(2j + 1) \times (2j + 1)$, *even order magic square* $(4j) \times (4j)$, *and even order magic square* $(4j + 2) \times (4j + 2)$ *where* $j = 1,2, \dots, m$.

Keywords: *Algorithm, Integer, Magic Square*

PENDAHULUAN

Matematika dikenal dengan sebutan induk dari ilmu-ilmu yang berkembang. Sadar atau tidak sadar, matematika digunakan hampir di setiap aspek kehidupan (Gafoor & Kurukkan, 2015). Matematika biasanya diperkenalkan di awal jenjang pendidikan sekolah yang merupakan mata pelajaran umum dan merupakan bagian mendasar dari kurikulum sekolah. Sebagian besar siswa menganggap matematika adalah ilmu yang sulit dipahami bahkan ada yang tidak suka dan merasa bosan dengan matematika (Kislenko, Grevholm, & Lepik, 2007). Pada umumnya hal tersebut dirasakan sejak bangku sekolah dasar sehingga pemikiran tersebut memengaruhi alam bawah sadar hingga dewasa kelak (Kunwar, 2020). Selama ribuan tahun, (*puzzles*) teka-teki matematika seperti persegi ajaib telah membangkitkan minat anak-anak dan orang dewasa untuk diselesaikan (Namakshi, Bhattacharyya, Starkey, & Linker, 2015).

Pada zaman prasejarah telah ditemukan kajian dan contoh-contoh persegi ajaib dalam literatur China. Ahli nujum Arab juga menggunakan persegi ajaib dalam perhitungan horoskop di abad kesembilan (Andrews, 1917). Dalam turunan ilmu matematika, persegi ajaib termasuk ke dalam masalah kombinatorika yang berkaitan dengan memilih objek dari sekumpulan pengaturan objek (Wallis & George, 2017). Ada dua jenis masalah yang umum terjadi, yaitu adanya pengaturan dan klasifikasi pengaturan. Jika jumlah pengaturan untuk suatu masalah tertentu kecil, pengaturan tersebut dapat dibuat daftar. Sedangkan jumlah pengaturan yang cukup banyak akan lebih baik tidak dibuat daftar terlebih dahulu. Menurut Berndt & Brualdi (2010), ada dua masalah kombinatorial yang sering terjadi, di antaranya mengetahui proses pengaturan dan konstruksi penataan yang optimal.

Permasalahan persegi ajaib adalah bagaimana cara menyusun angka dalam kotak persegi sedemikian sehingga baris, kolom dan sepanjang diagonal berjumlah sama (Sallows, 2011). Persegi ajaib terbagi menjadi tiga, yaitu: persegi ajaib ordo ganjil, persegi ajaib ordo genap (hasil bagi dua yang berjumlah ganjil) dan persegi ajaib ordo genap kelipatan empat (Gupta, 2019). Penelitian pengonstruksian persegi ajaib telah dilakukan oleh Sesiano (2019), Ashhab (2012), Baksalary, Trenkler, & Trenkler (2018) dan penerapan persegi ajaib ke dalam masalah tertentu telah dilakukan oleh Sahni & Ojha (2012), Dean (2017) dan Adetokunbo, Al-Shuhail, & Al-Dossary (2016). Penelitian ini membahas pengonstruksian persegi ajaib dengan sejumlah m^2 bilangan bulat berurutan yang diatur untuk menemukan pola tertentu dan algoritma persegi ajaib kemudian diimplementasikan ke dalam bahasa *Python* untuk menyelesaikan ordo yang besar.

METODE

Pengonstruksian persegi ajaib ordo $m \times m$ dibagi menjadi ordo ganjil $(2j + 1) \times (2j + 1)$ dan genap kemudian ordo genap dibagi menjadi ordo $(4j) \times (4j)$ dan ordo $(4j + 2) \times (4j + 2)$ dengan $j = 1,2, \dots\ , m$. Tahapan yang dilakukan adalah sebagai berikut:

1. Merumuskan masalah menentukan pengaturan pembentukan persegi ajaib ordo ganjil dan genap secara manual
2. Mengonstruksi algoritma persegi ajaib ordo ganjil dan genap
3. Melakukan uji kelayakan algoritma masing-masing ordo hingga ordo yang besar menggunakan *software Python*
4. Membuat program pengonstruksian persegi ajaib ordo $m \times m$ dengan mengombinasikan algoritma yang telah teruji menggunakan *software Python*.

HASIL DAN PEMBAHASAN

Deskripsi Masalah

Persegi ajaib ordo $m \times m$ dapat dimisalkan sebagai wujud matriks persegi yang berordo $m \times m$. Ordo atau ukuran dalam suatu matriks harus berupa bilangan bulat (*integer*) positif. Dalam rentang m bilangan bulat positif maka dapat dibagi menjadi bilangan ganjil dan genap. Ganjil dan genap dijadikan acuan sebagai penyelesaian persegi ajaib ordo $m \times m$. Pada penelitian ini persegi ajaib ordo $m \times m$ merupakan gabungan persegi ajaib ordo ganjil dan genap yang entri matriksnya bilangan bulat yang berurutan. Persegi ajaib ordo $m \times m$ adalah kotak-kotak berukuran $m \times m$ yang berisi angka berbeda kemudian diatur dan disusun sehingga baris, kolom dan diagonal utama berjumlah sama. Persegi ajaib ordo genap adalah suatu matriks yang mempunyai baris dan kolom berjumlah genap dan persegi ajaib ordo ganjil adalah suatu matriks yang mempunyai baris dan kolom berjumlah ganjil.

Konstruksi Persegi Ajaib

Pengonstruksian persegi ajaib dimulai dari ordo 3×3 dan seterusnya karena untuk ordo selainnya pengonstruksian persegi ajaib hanya dapat diselesaikan dengan angka yang sama dan tidak berurutan. Pengonstruksian persegi ajaib pada umumnya mempunyai banyak cara dan setiap ordo mempunyai pola pengonstruksian yang berbeda. Pada penelitian ini penyelesaian persegi ajaib ordo $m \times m$ dibagi menjadi ordo ganjil dan genap kemudian ordo genap dibagi menjadi ordo $(4j) \times (4j)$ dan ordo $(4j + 2) \times (4j + 2)$ dengan $j = 1,2, \ldots\ , m$. Pola pengonstruksian persegi ajaib ordo $m \times m$ didasari oleh ordo terkecil dari masing-masing ordo. Untuk menyelesaikan persegi ajaib ordo ganjil mengikuti pola pengonstrukian persegi ajaib ordo 3×3, persegi ajaib ordo genap $(4j) \times (4j)$ mengikuti pola pengonstruksian persegi ajaib ordo 4×4, persegi ajaib ordo genap $(4j + 2) \times (4j + 2)$ mengikuti pola pengonstruksian persegi ajaib ordo 6×6. Jumlah baris, kolom dan diagonal utama persegi ajaib dapat ditentukan dengan persamaan berikut ini

$$S = \frac{m}{2}(2w + (m^2 - 1)d)$$

dengan

$m =$ ordo persegi ajaib
$w =$ angka awal
$d =$ selisih setiap angka (berurutan) $= 1$
$S =$ jumlah baris, kolom dan diagonal utama.

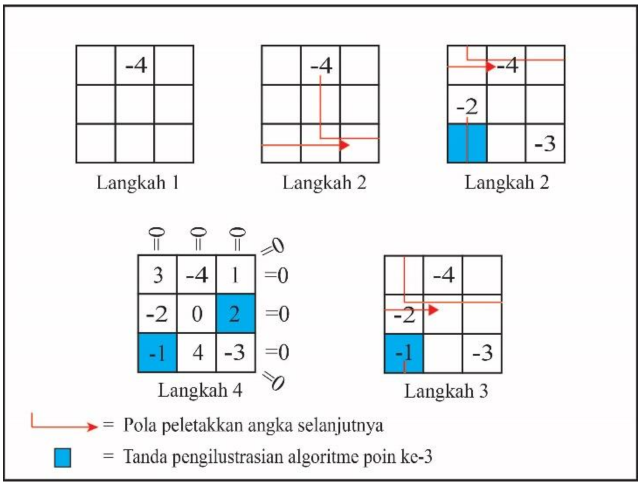
Berikut ini algoritma penyusunan persegi ajaib dan ditampilkan gambar ilustrasi di setiap langkah untuk menjelaskan algoritma lebih lanjut.

Algoritma Persegi Ajaib Ordo Ganjil $(\mathbf{2}\boldsymbol{j} + \mathbf{1}) \times (\mathbf{2}\boldsymbol{j} + \mathbf{1})$

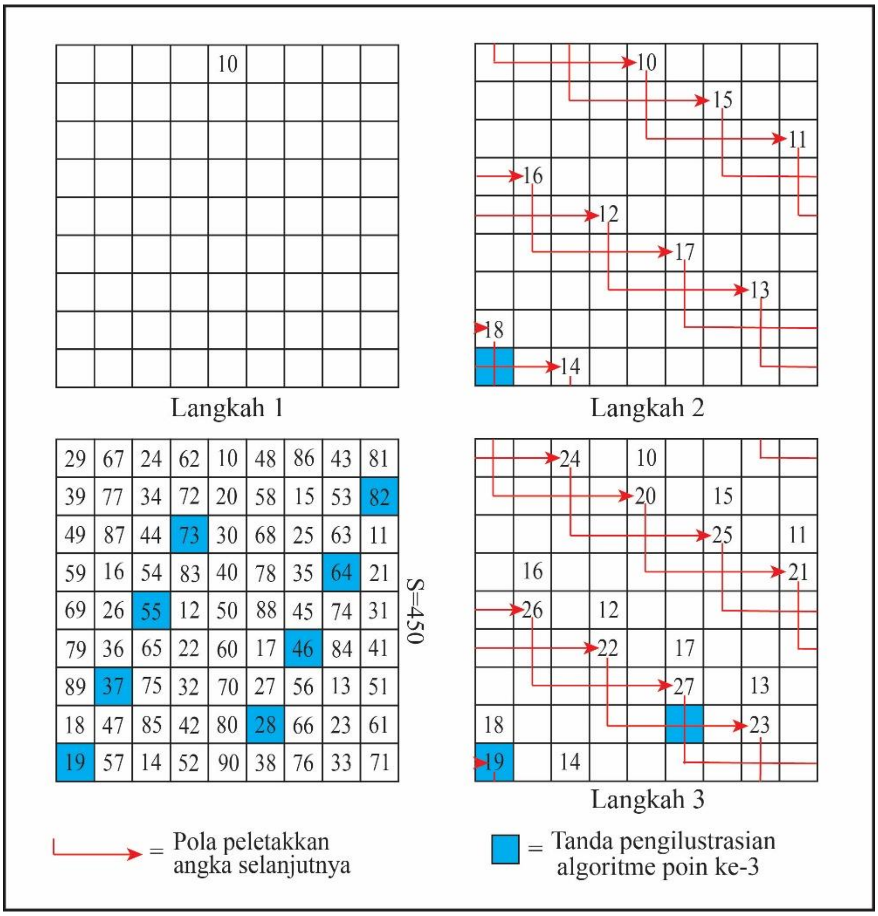
Penyusunan dan pengaturan angka pada persegi ajaib berdasarkan pola tertentu. Dalam hal ini jika pola penyusunan melewati batas terakhir baris atau kolom matriks maka dapat diteruskan ke baris atau kolom awal. Berikut algoritma persegi ajaib ordo ganjil dengan entri bilangan bulat yang berurutan:

1. Letakkan angka awal (bilangan bulat) pada atas-tengah matriks berordo ganjil
2. Letakkan angka selanjutnya berjarak bawah-bawah-kanan-kanan-kanan-kanan secara berkala, dan
3. Jika entri matriks sudah terisi angka maka letakkan di bawah angka tersebut, kemudian dilanjutkan ke langkah kedua
4. Lakukan langkah kedua dan ketiga hingga terbentuk persegi ajaib ordo ganjil.

Untuk menjelaskan algoritma di atas, berikut ilustrasi algoritma persegi ajaib ordo ganjil $(2j + 1) \times (2j + 1)$ dengan $j = 1$ dan $j = 4$.



Gambar 1. Ilustrasi Algoritma Persegi Ajaib Ordo 3 × 3 dengan Angka Awal – 4



Gambar 2. Ilustrasi Algoritma Persegi Ajaib Ordo 9 × 9 dengan Angka Awal 10

Berdasarkan Gambar 1 dan 2, karena baris, kolom dan diagonal utama berjumlah sama. Maka pola pengonstruksian ini berlaku untuk ordo ganjil. Persegi ajaib dengan ordo besar diselesaikan menggunakan *Python.* Berikut *pseudocode* persegi ajaib ordo ganjil:

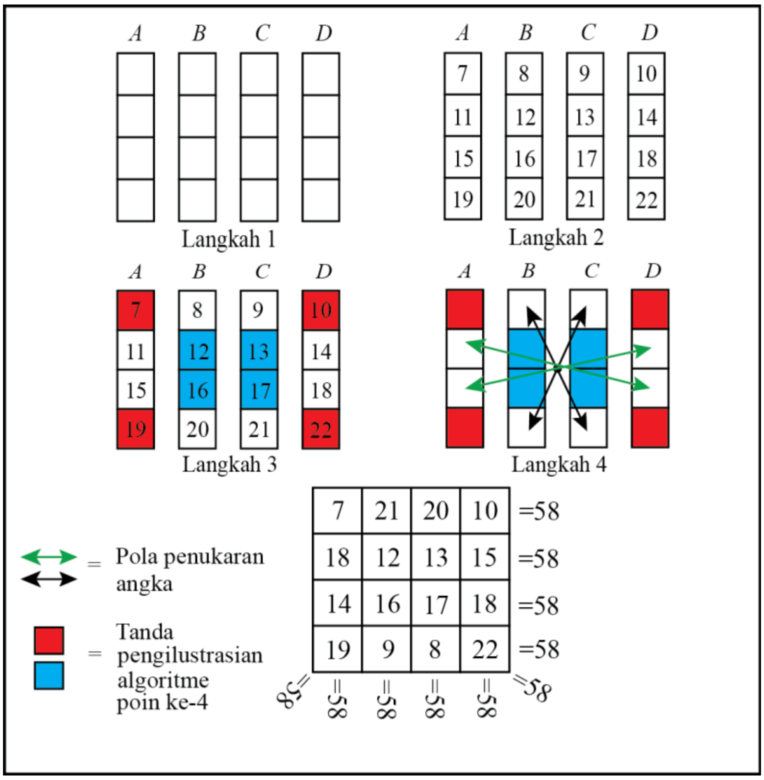
```
Program persegi ajaib ordo ganjil (2j + 1) × (2j + 1)
Inisialisasi
input sekumpulan bilangan bulat dengan angka awal tertentu,
beda dan persamaan jumlah baris, kolom dan diagonal utama
Proses
konstruksikan matriks berordo ganjil
isi angka awal di pusat atas matriks
while angka awal ≤ ordo matriks dikuadratkan dikali beda
     angka selanjutnya = angka awal + beda
     isi angka selanjutnya dengan jarak bawah-bawah-kanan-
     kanan-kanan-kanan
     if entri matriks sudah terisi angka
          isi angka selanjutnya di bawah angka sebelumnya
     elseif
          lanjutkan iterasi
     endif
endwhile
cetak matriks dan persamaan jumlah baris, kolom dan diagonal
utama.
```

Algoritma Persegi Ajaib Ordo Genap $(4j) \times (4j)$

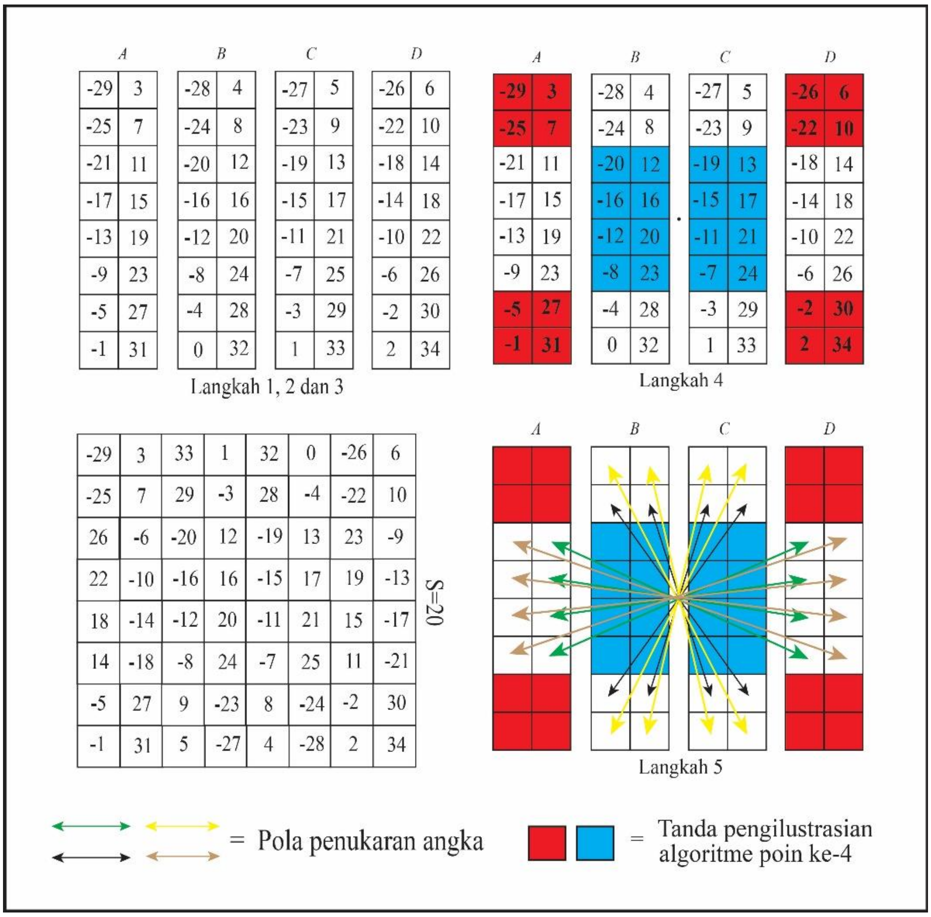
Pengonstruksian persegi ajaib ordo genap terbagi menjadi dua karena dalam bilangan bulat terdapat bilangan genap yang kelipatan empat dan hasil bagi dua yang berjumlah ganjil. Berikut algoritma persegi ajaib ordo genap $(4j) \times (4j)$ dengan entri bilangan bulat yang berurutan:

1. Partisi kolom matriks ordo genap $(4j) \times (4j)$ menjadi empat matriks kolom, misal
$$\boldsymbol{G} = [A \quad B \quad C \quad D]$$
2. Letakkan angka awal pada baris dan kolom pertama matriks A, dilanjutkan ke matriks B,C dan D kemudian dilanjutkan ke baris kedua kolom pertama pada matriks A dilanjutkan ke matriks B,C, dan D secara berulang hingga kolom pertama pada matriks A,B,C dan D terisi
3. Lakukan langkah kedua untuk mengisi kolom selanjutnya jika ada pada matriks A,B,C dan D (untuk $j \geq 2$)
4. Berikan tanda pada entri di setiap area sudut matriks dengan ukuran $\left(\frac{1}{4}(4j)\right) \times \left(\frac{1}{4}(4j)\right)$ dan $\left(\frac{1}{2}(4j)\right) \times \left(\frac{1}{2}(4j)\right)$ pada area pusat matriks
5. Tukar entri yang tidak diberi tanda dengan pasangan entri di seberangnya secara silang dan teratur, kemudian gabungkan sehingga terbentuk persegi ajaib ordo genap $(4j) \times (4j)$.

Untuk menjelaskan algoritma di atas, berikut ilustrasi algoritma persegi ajaib ordo genap $(4j) \times (4j)$ dengan $j = 1$ dan $j = 2$.



Gambar 3. Ilustrasi Algoritma Persegi Ajaib Ordo 4×4 dengan Angka Awal 7



Gambar 4. Ilustrasi Algoritma Persegi Ajaib Ordo 8×8 dengan Angka Awal -29

Dipilih dua pengonstruksian persegi ajaib ordo genap pada Gambar 3 dan 4 untuk membuktikan algoritma ini berlaku untuk persegi ajaib ordo genap kelipatan empat. Pengonstruksian ordo yang lebih besar dibantu dengan *software Python*. Berikut *pseudocode* persegi ajaib ordo genap $(4j) \times (4j)$:

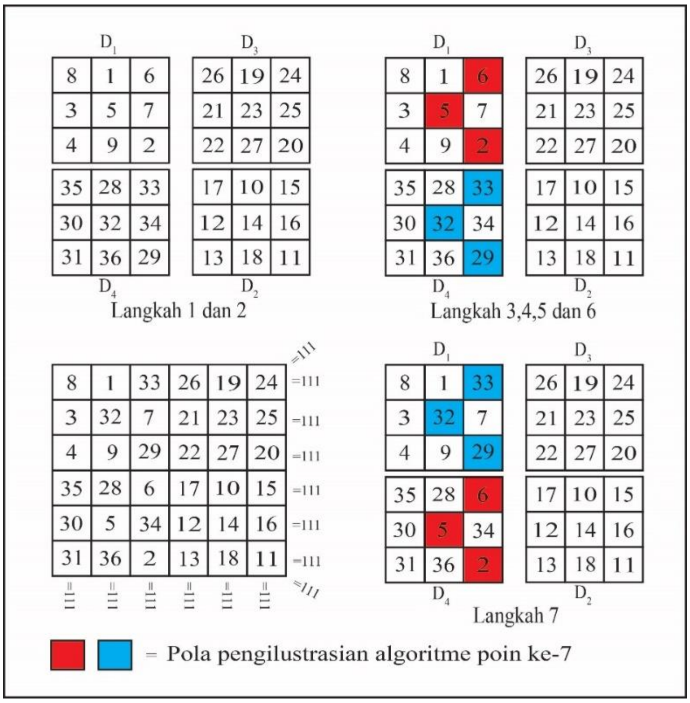
```
Program persegi ajaib ordo genap (4j) × (4j)
Inisialisasi
input ordo matriks, persamaan jumlah baris, kolom dan diagonal
utama, beda dan array matriks
Proses
while angka awal ≤ ordo matriks dikuadratkan dikali beda
     angka selanjutnya = angka awal + beda
     isi angka selanjutnya ke dalam array matriks
endwhile
reshape array matriks menjadi ordo genap (4j) × (4j) yang
dipartisi menjadi 1/4 bagian dari jumlah kolom
tukar beberapa entri matriks yang diperlukan
cetak matriks dan persamaan jumlah baris, kolom dan diagonal
utama.
```

Algoritma Persegi Ajaib Ordo Genap $(4j+2) \times (4j+2)$

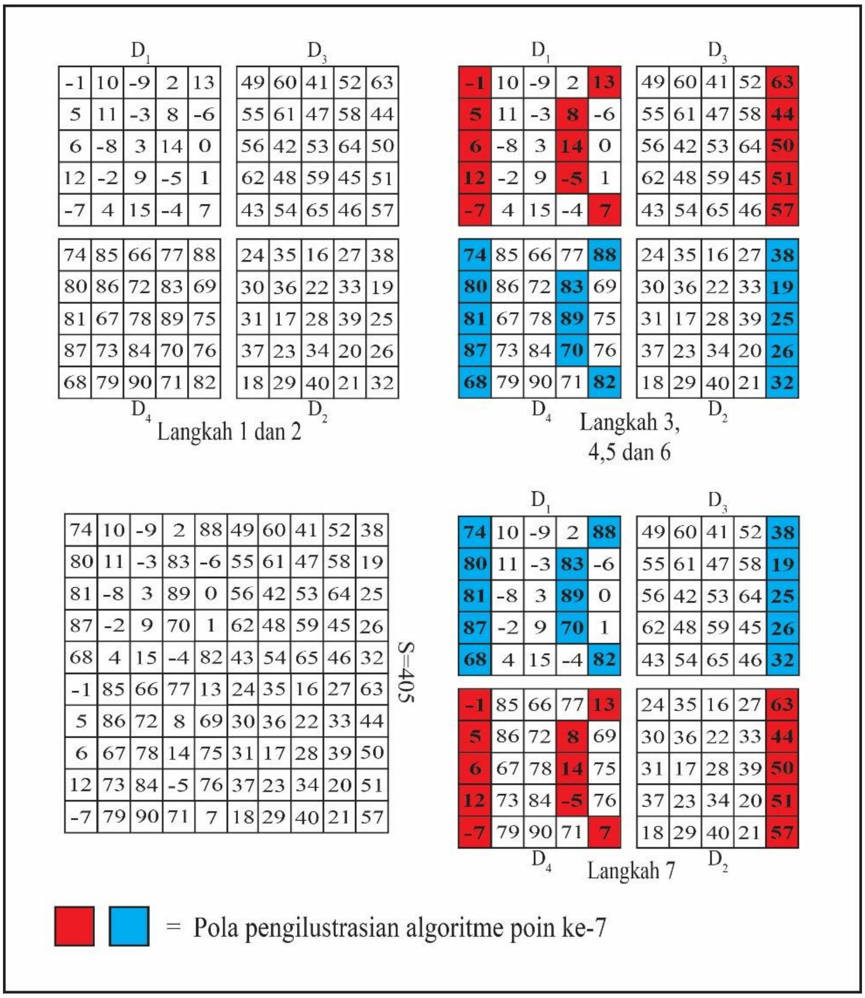
Pengonstruksian algoritma persegi ajaib ordo genap $(4j+2) \times (4j+2)$ merupakan gabungan persegi ajaib ordo ganjil yang berurutan kemudian disusun dengan aturan tertentu. Berikut algoritma persegi ajaib ordo genap $(4j+2) \times (4j+2)$ dengan entri bilangan bulat yang berurutan:

1. Partisi kolom matriks ordo genap $(4j+2) \times (4j+2)$ menjadi matriks 2×2 yang diatur sedemikian sehingga
$$\boldsymbol{E} = \begin{bmatrix} \boldsymbol{D_1} & \boldsymbol{D_3} \\ \boldsymbol{D_4} & \boldsymbol{D_2} \end{bmatrix}$$
2. $\boldsymbol{D_1}, \boldsymbol{D_2}, \boldsymbol{D_3}$, dan $\boldsymbol{D_4}$ adalah persegi ajaib ordo ganjil yang disusun menggunakan algoritma persegi ajaib ordo ganjil secara berurutan
3. Berikan tanda pada entri sudut atas dan bawah kolom ke t matriks $\boldsymbol{D_1}$ dan kolom $t-1$ kecuali baris awal dan akhir pada matriks $\boldsymbol{D_1}$ dengan t adalah kolom terakhir matriks $\boldsymbol{D_1}$
4. Lakukan langkah ketiga untuk matriks $\boldsymbol{D_4}$
5. Berikan tanda pada $\left(\frac{1}{4}\left((4j+2)-6\right)\right)$ kolom pertama matriks $\boldsymbol{D_1}$ dan $\boldsymbol{D_4}$
6. Berikan tanda pada $\left(\frac{1}{4}\left((4j+2)-6\right)\right)$ kolom terakhir matriks $\boldsymbol{D_3}$ dan $\boldsymbol{D_2}$
7. Tukar area yang diberi tanda, matriks $\boldsymbol{D_1}$ dengan matriks $\boldsymbol{D_4}$ dan matriks $\boldsymbol{D_3}$ dengan matriks $\boldsymbol{D_2}$ kemudian gabungkan sehingga terbentuk persegi ajaib ordo genap $(4j+2) \times (4j+2)$.

Untuk menjelaskan algoritma di atas, berikut ilustrasi algoritma persegi ajaib ordo genap $(4j+2) \times (4j+2)$ dengan $j = 1$ dan $j = 2$.



Gambar 5. Ilustrasi Algoritma Persegi Ajaib Ordo 6×6 dengan Angka Awal 1



Gambar 6. Ilustrasi Algoritma Persegi Ajaib Ordo 10×10 dengan Angka Awal -9

Gambar 5 dan 6 membuktikan bahwa baris, kolom dan diagonal utama berjumlah sama. Akibatnya algoritma ini berlaku untuk persegi ajaib ordo genap $(4j+2)\times(4j+2)$ untuk $j=1,2,\dots,m$. Untuk $j>3$, pengonstruksian persegi ajaib dibantu dengan *software Python*. Berikut *pseudocode* persegi ajaib ordo genap $(4j+2)\times(4j+2)$:

```
Program persegi ajaib ordo genap (4j + 2) × (4j + 2)
Insialisasi
input sekumpulan bilangan bulat dengan angka awal tertentu,
beda dan persamaan jumlah baris, kolom dan diagonal utama
def matriks_1d
    konstruksikan matriks berordo ganjil
    isi angka awal di pusat atas matriks
    while angka awal ≤ ordo matriks dikuadratkan dikali beda
        angka selanjutnya = angka awal + beda
        isi angka selanjutnya dengan jarak bawah-bawah-
        kanan-kanan-kanan-kanan
        if entri matriks sudah terisi angka
            isi angka selanjutnya di bawah angka sebelumnya
        elseif
            lanjutkan iterasi
        endif
    endwhile
    return matriks_1d
enddef
def matriks_2d
    konstruksikan matriks berordo ganjil
    isi angka awal melanjutkan angka terakhir matriks_1d di
    pusat atas matriks
    while angka awal ≤ ordo matriks dikuadratkan dikali beda
        angka selanjutnya = angka awal + beda
        isi angka selanjutnya dengan jarak bawah-bawah-
        kanan-kanan-kanan-kanan
        if entri matriks sudah terisi angka
            isi angka selanjutnya di bawah angka sebelumnya
        elseif
            lanjutkan iterasi
        endif
    endwhile
    return matriks_2d
enddef
def matriks_3d
    konstruksikan matriks berordo ganjil
    isi angka awal melanjutkan angka terakhir matriks_2d di
    pusat atas matriks
    while angka awal ≤ ordo matriks dikuadratkan dikali beda
        angka selanjutnya = angka awal + beda
        isi angka selanjutnya dengan jarak bawah-bawah-
        kanan-kanan-kanan-kanan
        if entri matriks sudah terisi angka
            isi angka selanjutnya di bawah angka sebelumnya
        elseif
            lanjutkan iterasi
```

```
        endif
    endwhile
    return matriks_3d
enddef
def matriks_4d
    konstruksikan matriks berordo ganjil
    isi angka awal melanjutkan angka terakhir matriks_3d di
    pusat atas matriks
    while angka awal ≤ ordo matriks dikuadratkan dikali beda
        angka selanjutnya = angka awal + beda
        isi angka selanjutnya dengan jarak bawah-bawah-
        kanan-kanan-kanan-kanan
        if entri matriks sudah terisi angka
            isi angka selanjutnya di bawah angka sebelumnya
        elseif
            lanjutkan iterasi
        endif
    endwhile
    return matriks_4d
enddef
gabungkan matriks_1d dengan 4d dan matriks_2d dengan 3d secara
vertikal
def penukaran entri di tengah
    tukar entri di sudut kanan-atas dan kanan-bawah pada
    kolom ke t matriks_1d dengan matriks_4d
    tukar kolom ke (t − 1) pada matriks_1d kecuali entri paling
    atas dan bawah dengan matriks_4d
    # t = kolom matriks terakhir
enddef
def penukaran entri di pinggir
    tukar (1/4((4j + 2) − 6)) kolom pertama pada matriks_1d dengan
    matriks_4d dan tukar (1/4((4j + 2) − 6)) kolom terakhir pada
    matriks_2d dengan matriks_3d
enddef
if ordo matriks = 6
    do def penukaran entri di tengah
elseif
    do def penukaran entri di tengah dan pinggir
endif
```

KESIMPULAN

Pengonstruksian persegi ajaib didasari dengan pembentukan pola dan algoritma. Persegi ajaib dengan entri bilangan bulat dibagi menjadi tiga kasus yaitu; persegi ajaib ordo ganjil $(2j+1)\times(2j+1)$, persegi ajaib ordo genap $(4j)\times(4j)$, dan persegi ajaib ordo genap $(4j+2)\times(4j+2)$ dengan $j=1,2,\dots,m$. Pola dan algoritma persegi ajaib untuk $j=1$ pada masing-masing kasus dijadikan sebagai dasar pengonstruksian persegi ajaib untuk $j=2,3,4,\dots,m$.

DAFTAR RUJUKAN

Adetokunbo, P., Al-Shuhail, A. A., & Al-Dossary, S. (2016). 3D Seismic Edge Detection Using Magic Squares and Cubes. *Interpretation*, *4*(3), T271–T280.

Andrews, W. S. (1917). *Magic Squares and Cubes*. USA: Open Court Publishing.

Ashhab, S. (2012). Special Magic Squares of Order Six and Eight. *International Journal of Digital Information and Wireless Communications*, *1*(4), 769–781.

Baksalary, O. M., Trenkler, D., & Trenkler, G. (2018). On Most Perfect Magic Square of Order Four. *Linear and Multilinear Algebra*, 1–13.

Berndt, B. C., & Brualdi, R. A. (2010). *Introductory Combinatorics.* Beijing: China Machine Press.

Dean, A. (2017). The Magic Square of Olocin Ozzaniugnas. *Early Music*, *45*(4), 599–611.

Gafoor, K. A., & Kurukkan, A. (2015). Why High School Students Feel Mathematics Difficult? An Exploration of Affective Beliefs. *UGC Sponsored National Seminar on Pedagogy of Teacher Education Trends and Challenges*, 1–6.

Gupta, B. (2019). Algorithm for Doubly Even Magic Square. *International Advanced Research Journal in Science, Engineering and Technology*, *6*(3), 112–117.

Kislenko, K., Grevholm, B., & Lepik, M. (2007). "Mathematics is Important but Boring": Students Beliefs and Attitudes Towards Mathematics. *Mathematics Educational Research Journal.*, *11*(1), 39–53.

Kunwar, R. (2020). Math mania : Meaning , Problems and Ways of Effective Teaching and Learning Mathematics at Basic Level Education in Nepal. *International Journal of Science and Research*, *9*(8), 1136–1141.

Namakshi, N., Bhattacharyya, S., Starkey, C., & Linker, J. M. (2015). Mystical Magic Square. *National Council of Teachers of Mathematics*, *20*(6), 372–377.

Sahni, M., & Ojha, D. B. (2012). Magic Square and Cryptography. *Journal of Global Research in Computer Science*, *3*(12), 15–17.

Sallows, L. (2011). Geometric Magic Squares. *The Mathematical Intelligencer*, *33*(4), 25–31. https://doi.org/10.1007/s00283-011-9229-0

Sesiano, J. (2019). *Magic Squares: Their History and Construction from Ancient Times to AD 1600*. Geneva: Springer.

Wallis, W. D., & George, J. C. (2017). *Introduction to Combinatorics*. London: CRC Press.