

# Analisis Data Menggunakan Algoritma Pararel Matriks Shuffle Transpose Menggunakan Bahasa Pemrograman C#

Danang Aji Pangestu<sup>1</sup>, Mikhael Kristian<sup>2</sup>, Aldy Febriyandani<sup>3</sup>, Aris Gunaryati<sup>4</sup>

Program Studi Informatika, Universitas Nasional Jakarta

Email: <sup>1</sup>nangjipang@gmail.com, <sup>2</sup>kristianmikhael667@gmail.com, <sup>3</sup>aldyfebriyandani@gmail.com,

<sup>4</sup>aris.gunaryati@civitas.unas.ac.id

**Abstrak:** Tak jarang kira dihadapkan dengan sebuah data, dalam menyelesaikan sebuah data kita harus memakai sebuah metode agar permasalahan dapat selesai dengan baik, salah satu metodenya yaitu matriks shuffle transpose. Transpose Shuffle adalah digunakan untuk interkoneksi perfect shuffle dengan  $n^2$  pada tiap-tiap processor dan akan menghasilkan data berupa transpose matriks yang akan merubah elemen baris dan kolom. Sama seperti transpose pada umumnya, lambang tanda seperti  $(A')$  atau biasa digunakan adalah  $(A^T)$ .

**Kata Kunci:** Matrix, Transpose Method, Transpose Shuffle

**Abstract:** Not infrequently think faced with a data, in completing a data we must use a method so that the problem can be solved properly, one of the methods is the shuffle transpose matrix. Transpose Shuffle is used to interconnect perfect shuffle with  $n^2$  on each processor and will produce data in the form of a transpose matrix that will change baris and column elements. Just like transpose in general, a symbol like  $(A')$  or commonly used is  $(A^T)$ .

**Keywords:** Matrix, Transpose Method, Transpose Shuffle

## PENDAHULUAN

Didalam Kehidupan Sehari-Hari Kita Tak Asing Dengan Yang Namanya Data, Data Merupakan Sebuah Informasi Mentah Yang Belum Diolah. Data Biasanya Digunakan Untuk Menjabarkan Sebuah Permasalahan. Penjabaran Data Tersebut Dapat Digunakan Dalam Sebuah Matriks. Matriks Dapat Digunakan Untuk Menyelesaikan Sebuah Permasalahan Secara Pararel Digunakan Untuk Interkoneksi Perfect Shuffle Dengan  $N^2$  [1].

Penyelesaian algoritma tersebut dapat diselesaikan dengan menggunakan bahasa pemrograman yaitu C#. C# merupakan bahasa yang bersifat oriented programing yang dapat dijadikan sebuah aplikasi mobile, desktop, dan lain sebagainya.

Penulisan ini bertujuan membahas algoritma pararel tentang metode matriks shuffle transpose dan bagaimana cara penerapannya dalam menggunakan bahasa pemrograman c#.

Penyelesaian metode ini dapat diselesaikan dengan cara membagi dan memindahkn data kedalam sebuah prosesor sebelum dilakukan penukaran data atau transpose[5].

Pada umumnya Transpose Shuffle di anggap kecil karena jumlah prosesor kuadratik, Elemen yang berbeda dapat mentranpose sekumpulan angka yang disusun secara baris atau kolom. Geometri yang berbeda dpt men-transpose sebuah matriks dalam waktu logaritmik [2].

## METODE PENYELESAIAN

### Shuffle Transpose

Kecepatan dari Mesh transpose hanya linier, dianggap kecil karena jumlah prosesor kuadratik. geometri yg berbeda dpt men-transpose matriks dalam waktu logaritmik.

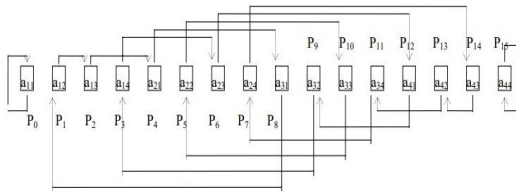
$$n = \text{Element pangkat } q$$

Element pangkat q merupakan bilangan akar kuadrat, artinya syarat dari matriks transpose

shuffle harus berordo sama antara baris dan kolom.

Dalam menyelesaikan Tranpose Shuffle berdasarkan data yang telah kita peroleh, memperoleh dengan digunakan interkoneksi perfect shuffle dgn  $n^2$ ; prosesor:  $p_0, p_1, \dots, p_{n-1}$ .

$a_{ij}$  disimpan pada awalnya tersimpan di prosesor  $P_k$ , dimana  $k = 2q(i-1) + (j-1)$



Gambar 1. Perpindahan elemen pada prosesor.

Indeks prosesor  $k$  terdiri dari  $2q$  bit.  $q$  menunjukkan sebagian besar bit menyatakan  $i-1$ , dan  $q$  sebagian kecil bit menyatakan  $j-1$ . Setelah elemen  $q$  shuffle mengalami pergeseran ruas ke kiri, elemen yang sebelumnya berada pada  $P_k$  akan berada diprosesor yang indeksnya adalah:

$$s = 2q(j-1) + (i-1)$$

**Algoritma Penyelesaian**

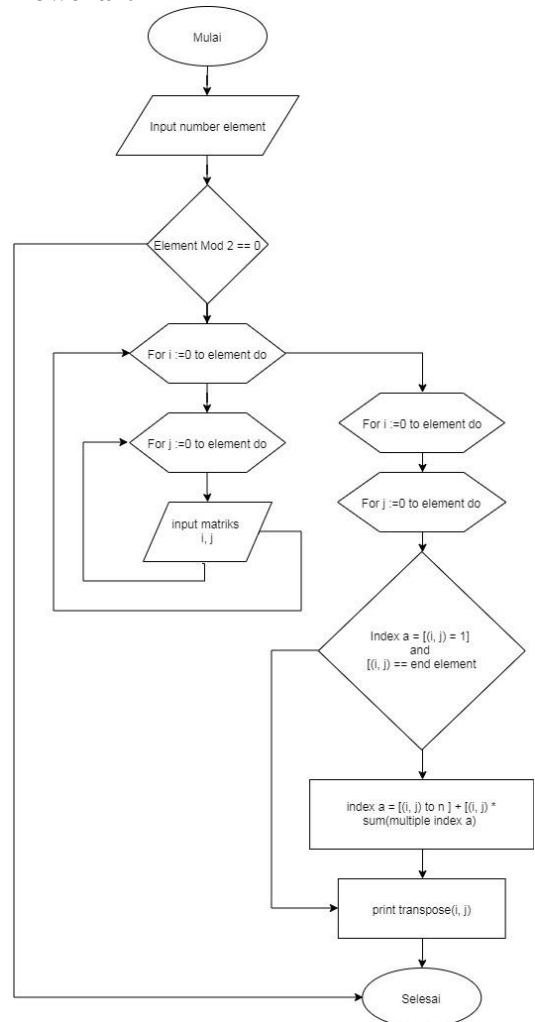
```

for i=1 to q do
    for k=1 to 2q-2 do in parallel
        Pk sends the element of A
        currently holds to P2
        kmod(2q-1)
    end for
end for.
    
```

Analisis:

- Iterasi:  $q$  waktu konstan
- Waktu:  $t(n) = O(\log n)$
- $p(n) = n^2$ ;  $c(n) = O(n^2 \log n) \rightarrow$  tidak optimal
- Tetapi interkoneksi shuffle lebih cepat jika dibandingkan dengan  $m$

**Flowchart**



Gambar 2. Flowchart Algoritma shuffle transpose.

**HASIL DAN PEMBAHASAN**

**Studi Kasus**

Diberikan sebuah data sebanyak 16. Dimana data tersebut yaitu [1, 3, 5, 8, 7, 9, 4, 1, 2, 3, 4, 4, 1, 1, 5, 3]. Dimana data tersebut dilakukan secara matriks shuffle transpose.

Penyajian data :

Before Transpormation :

1	3	5	8	p0	p1	p2	p3
7	9	4	1	p4	p5	p6	p7
2	3	4	4	p8	p9	p10	p11
1	1	5	3	p12	p13	p14	p15
	(data)				(Prosesor)		

Jalannya proses :

a11: P0 → P0 → P0 = 1  
 a12: P1 → P2 → P4 = 7  
 a13: P2 → P4 → P8 = 2  
 a14: P3 → P6 → P12 = 1  
 a21: P4 → P8 → P1 = 3  
 a22: P5 → P10 → P5 = 9  
 a23: P6 → P12 → P9 = 3  
 a24: P7 → P14 → P13 = 1  
 a31: P8 → P1 → P2 = 5  
 a32: P9 → P3 → P6 = 4  
 a33: P10 → P5 → P10 = 4  
 a34: P11 → P7 → P14 = 5  
 a41: P12 → P9 → P3 = 8  
 a42: P13 → P11 → P7 = 1  
 a43: P14 → P13 → P11 = 4  
 a44: P15 → P15 → P15 = 3

After Transformation :

p0	p1	p2	p3	1	7	2	1
p4	p5	p6	p7	3	9	3	1
p8	p9	p10	p11	5	4	4	5
p12	p13	p14	p15	8	1	4	3
(Prosesor)				(Data)			

Dapat dilihat dari penyelesaian tersebut, data tersebut mengalami pertukaran antara  $a_{ij}$  dengan  $a_{ji}$ .

### Implementasi Dalam Program

```
file:///D:/danang/uas_ap/uas_ap/bin/Debug/uas_ap.EXE
MASUKAN MATRIKS :
4
baris ke-[1], kolom ke-[1] :1
baris ke-[1], kolom ke-[2] :3
baris ke-[1], kolom ke-[3] :5
baris ke-[1], kolom ke-[4] :8
baris ke-[2], kolom ke-[1] :7
baris ke-[2], kolom ke-[2] :9
baris ke-[2], kolom ke-[3] :4
baris ke-[2], kolom ke-[4] :1
baris ke-[3], kolom ke-[1] :2
baris ke-[3], kolom ke-[2] :3
baris ke-[3], kolom ke-[3] :4
baris ke-[3], kolom ke-[4] :4
```

Gambar 3. Input Data Kedalam Matriks

```
baris ke-[3], kolom ke-[3] :4
baris ke-[3], kolom ke-[4] :4
baris ke-[4], kolom ke-[1] :1
baris ke-[4], kolom ke-[2] :1
baris ke-[4], kolom ke-[3] :5
baris ke-[4], kolom ke-[4] :3
1      3      5      8
7      9      4      1
2      3      4      4
1      1      5      3
```

Gambar 4. Input Data Kedalam Matriks dan Matriks sebelum ditranspose

```
file:///D:/danang/uas_ap/uas_ap/bin/Debug/uas_ap.EXE
```

Jalannya Elemen

```
kolom ke-[1], baris ke-[1] : 1
kolom ke-[2], baris ke-[1] : 3
kolom ke-[3], baris ke-[1] : 5
kolom ke-[4], baris ke-[1] : 8
kolom ke-[1], baris ke-[2] : 7
kolom ke-[2], baris ke-[2] : 9
kolom ke-[3], baris ke-[2] : 4
kolom ke-[4], baris ke-[2] : 1
kolom ke-[1], baris ke-[3] : 2
kolom ke-[2], baris ke-[3] : 3
kolom ke-[3], baris ke-[3] : 4
kolom ke-[4], baris ke-[3] : 4
kolom ke-[1], baris ke-[4] : 1
kolom ke-[2], baris ke-[4] : 1
kolom ke-[3], baris ke-[4] : 5
kolom ke-[4], baris ke-[4] : 3
```

After Transformation

```
1      7      2      1
3      9      3      1
5      4      4      5
8      1      4      3
```

Gambar 5. Jalannya Elemen proses

## KESIMPULAN

Jadi dalam menentukan metode shuffle transpose matriks hal yang harus dilakukan adalah dengan cara menjumlahkan tiap prosesor pada kuadratik tersebut, kemudian elemen yang berbeda dari setiap baris maupun kolom dapat dilakukan dengan cara mentranspose sekumpulan angka yang disusun secara baris atau kolom. Geometri yang berbeda dapat men-transpose sebuah matriks dalam waktu logaritmik.

## DAFTAR PUSTAKA

- [1] Admin. (2018, September 10). *Matriks*. Retrieved from wikibooks.org: <https://id.wikibooks.org/wiki/Subjek:Matematika/Materi:Matriks>
- [2] Budiono, S. (2018). *Pengolahan Paralel*. Retrieved from docplayer: <https://docplayer.info/47708723-Pengolahan-paralel-pengolahan-paralel-ernastuti-1-58-ernastuti.html>
- [3] Damayanti, R. (2018). *Jurnal matriks*. Retrieved from docplayer.info: <https://docplayer.info/111922011-Jurnal-matriks-vol-1-no-2-2018.html>
- [4] Gunaryati, A. (2012, 12 12). *Operasi Matriks*. Retrieved from wordpress: <https://arisgunaryati.files.wordpress.com/2012/12/kuliah1011.pdf>
- [5] Rahim, R. (2010). S.kom. *Simplified memory bounded A\*pada proses penyelesaian operasi matriks*, 1.
- [6] Subaeki, B. (2017). S.Kom. *Implementasi Algoritma Fisher-Yates Shuffle Pada Aplikasi Multimedia Interaktif Untuk Pembelajaran Tenses Bahasa Inggris*, 2.