Mathematical Journal of Modelling and Forecasting
Vol. 3, No. 1, June 2025, pp. 1-7

The Introduction of Strassen's Algorithm and Application to 2^n Matrix Multiplication

Davina Anjelia[1,*], Meira Parma Dewi[2]

[1,2] Departement of Mathematics, Universitas Negeri Padang, Indonesia
[*] muthiaraputri04@gmail.com

Abstract. In matrix calculation operations, especially the process of square matrix multiplication, as the order of the matrix increases, the level of accuracy required also increases. Manual calculation is prone to errors and takes a long time, especially for large order matrices. These problems can be overcome by using the Strassen algorithm. Strassen's algorithm views a matrix as a 2×2 matrix because it has four elements. Square matrix multiplication using the Strassen algorithm can be an alternative solution because the Strassen algorithm only contains seven multiplication processes. So, applying the Strassen algorithm to square matrix multiplication will be an alternative in accelerating the multiplication process, especially for matrices of a large order. This research discusses how the Strassen algorithm is formed and its application to the square matrix multiplication of order 2^n. Strassen's algorithm is obtained by transforming the elements of the product matrix C. Algebraic identity transformation is done by applying the properties that apply to the calculation operation without changing the original value. Using Strassen's Algorithm in the square matrix multiplication process can be an alternative in accelerating the multiplication process because Strassen's algorithm summarises the multiplication process into seven steps, compared to multiplication in general, which requires eight steps.

Keywords: Identity Transformation, Strassen Algorithm, Square Matrix Multiplication, Web-Based Applications

1 Introduction

Algebra in the context of matrix multiplication can be done by multiplying each row element in the matrix A with each column element in the matrix B, then summing the row and column product [1]. Multiplication of matrices of a small order can be done by multiplying the two matrices manually, one by one. In multiplying two matrices, as the order of the two matrices increases, the accuracy required for the calculation also increases. So, in multiplying matrices of a large order, errors often occur in the calculation process. According to the Indonesian dictionary, an error is a mistake or wrongdoing [2]. Errors that can occur in the matrix calculation process cause a decrease in the accuracy of the multiplication results on matrices that have large orders. To overcome this problem, we can use the Strassen algorithm.

Strassen's algorithm was published by Volker Strassen in 1969. In the case of a square matrix of a large order, we can apply Strassen's algorithm to solve it. Strassen's algorithm views a matrix as a 2×2 matrix in the sense that it is composed of four elements. By using Strassen's algorithm, a matrix will be divided into several submatrix parts [3]. The Strassen algorithm summarises multiplying two square matrices into seven multiplication processes compared to the conventional method, which requires eight multiplication processes [4]. In the process of forming seven multiplication steps in the Strassen algorithm, an algebraic identity transformation is used. The transformation changes the equation's form without changing the original value. Where the transformation is applied to the 2×2 matrix multiplication is conventionally performed by utilizing the properties that apply to the matrix.

In the multiplication of matrices that have a small order, it can be easy to process the multiplication conventionally, one by one. However, in cases where the matrix has a significant or even substantial order, conventionally multiplying the matrix will take a lot of time. Strassen's algorithm can be used to perform the multiplication process, especially on matrices with substantial orders that can be found in real-life problems. The application of matrix multiplication is widely found in everyday life, such as image and image processing, computer graphics in 3D games, recommendation systems on social media, cryptography in data security, climate simulation and prediction, robotics and navigation systems, and many more. For this reason, research on matrix multiplication is significant in supporting its application in daily life.

2 Method

The algebraic identity transformation is used to find the origin of the Strassen algorithm. Algebra is a generalisation of various arithmetic ideas that deal with variables and unknown values and can be used to solve problems [5]. Transformation is the process of gradual change, up to the ultimate stage, by responding to the influence of external and internal elements [6]. Algebraic transformation refers to the technique of changing a form of mathematical expression to another form that is simpler or more appropriate to the needs of problem solving. This technique involves algebraic manipulations such as factoring, substitution, expansion, and matrix simplification.

The algebraic identity transformation uses the properties of multiplication to simplify algebraic expressions, which can be extended to the distributive properties of multiplication over addition. Consider the following property:

$$a(b+c) = ab + ac$$

$$a(b-c) = ab - ac$$

The two properties above will be used in the process of forming the Strassen algorithm, obtained by transforming the 2×2 matrix multiplication.

Strassen's algorithm also uses The divide-and-conquer technique in the matrix multiplication process. The divide and conquer technique involves three main steps, namely, divide to break the problem into several smaller and similar sub-problems, conquer to solve each sub-problem recursively, and finally combine, namely, combining the solutions of the sub-problems to form the solution of the original problem [7]. Divide and Conquer is often used in computer programming to break complex problems into smaller and more easily solved parts. Divide and Conquer will first divide $\frac{n}{2}$ of a problem and keep repeating it until it becomes the desired sub-problem. The basic concept is to break a significant problem into several smaller sub-problems, solve each sub-problem separately, and then combine the sub-problem solutions into a final solution to the larger problem [8]. The following illustrates the concepts of combine, conquer, and combine.

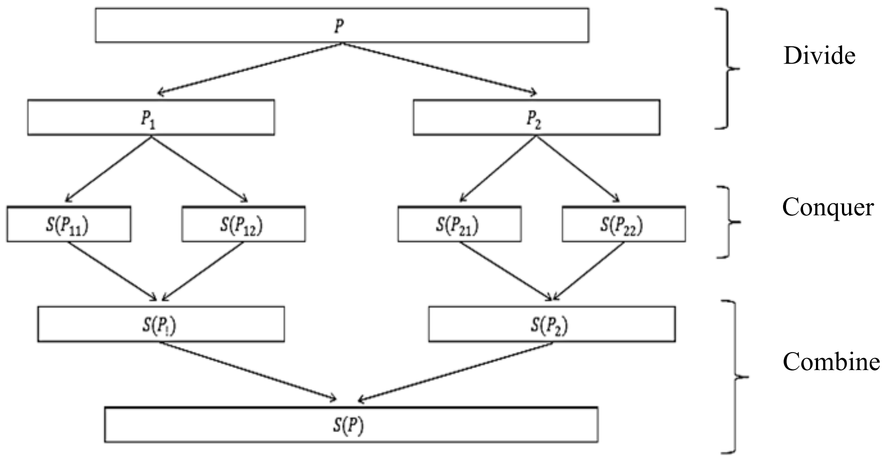

Fig 1. Illustration of the concept of divide, conquer and combine

First we will divide a problem (P) into sub-problems (P_i), if the sub-problems are still too large then the sub-problems will be divided into several sub-problems (P_{ij}), then the sub-problems will be solved ($S(P_{ij})$) which are then put together ($S(P_i)$) so that they become the solution of a complete problem (S(P)). then the algebraic identity transformation will be used in the formation of the strassen algorithm and the divide conquer concept is used in the application of the strassen algorithm to the multiplication of 2^n matrices.

The square matrix calculation operation, especially in the multiplication of two matrices with order 2^n, we can use the *Strassen* algorithm as an alternative to conventional methods. Multiplication of two square matrices, for example matrix A and matrix B with order $n \times n$, will produce a new matrix, i.e., matrix $(A \times B = C)$. Which is :

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix}; B = \begin{bmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{bmatrix}; C = \begin{bmatrix} C_{1,1} & C_{1,2} \\ C_{2,1} & C_{2,2} \end{bmatrix}$$

Elements in matrix C are calculated based on sub-matrix A and matrix B by using this approach $C[i,j] = \sum_{k=1}^{n} A[i,k] \cdot B[k,j]$. The Strassen algorithm will produce 7 multiplications. Suppose we have two square matrices A and B, each of size 2×2:

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix}; B = \begin{bmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{bmatrix};$$

The conventional multiplication result is :

$$C = A.B = \begin{bmatrix} C_{1,1} & C_{1,2} \\ C_{2,1} & C_{2,2} \end{bmatrix}$$

Strassen's algorithm divides the multiplication into seven matrices, i.e. :

a. $M_1 = (A_{1,1} + A_{2,2}) \cdot (B_{1,1} + B_{2,2})$

b. $M_2 = (A_{2,1} + A_{2,2}) \cdot B_{1,1}$

c. $M_3 = A_{1,1} \cdot (B_{1,2} - B_{2,2})$

d. $M_4 = A_{2,2} \cdot (B_{2,1} - B_{1,1})$

e. $M_5 = (A_{1,1} + A_{1,2}) \cdot B_{2,2}$

f. $M_6 = \left(A_{2,1} - A_{1,1}\right) \cdot \left(B_{1,1} + B_{1,2}\right)$

g. $M_7 = (A_{1,2} - A_{2,2}) \cdot (B_{2,1} + B_{2,2})$

The product C is given by:

1) $C_{1,1} = M_1 + M_4 - M_5 + M_7$;
2) $C_{1,2} = M_3 + M_5$;
3) $C_{2,1} = M_2 + M_4$;
4) $C_{2,2} = M_1 - M_2 + M_3 + M_6$.

3 Result and Discussion

3.1 The Process of Strassen's Algorithm

The multiplication process of a 2×2 square matrix generally requires 8 multiplication operations and 4 addition operations. Suppose $AB = [c_{ij}]$, then:

$$(A \times B)_{2\times 2} = [c_{ij}]$$

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \times \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{bmatrix}$$

$$(AB)_{2\times 2} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

The result of the conventional C result matrix above produces 8 multiplications. If the C result matrix is translated into a sub-matrix, then :

a. $M_1` = A_{11}B_{11}$

b. $M_2` = A_{12}B_{21}$

c. $M_3` = A_{11}B_{12}$

d. $M_4` = A_{12}B_{22}$

e. $M_5` = A_{21}B_{11}$

f. $M_6` = A_{22}B_{21}$

g. $M_7` = A_{21}B_{12}$

h. $M_8` = A_{22}B_{22}$

Furthermore, an algebraic identity transformation will be carried out to reduce the multiplication process. In performing the transformation, distributive and associative properties will be used.

1) $C_{11} = A_{11}B_{11} + A_{12}B_{21}$

$$A_{11}B_{11} + A_{12}B_{21} = A_{11}B_{11} + A_{11}B_{22} + A_{22}B_{11} + A_{22}B_{22} + A_{22}B_{21} - A_{22}B_{11} - A_{11}B_{22} - A_{12}B_{22} + A_{12}B_{21} + A_{12}B_{22} - A_{22}B_{21} - A_{22}B_{22}$$

$$A_{11}B_{11} + A_{12}B_{21} = (A_{11} + A_{22})(B_{11} + B_{22}) + A_{22}(B_{21} - B_{11}) - (A_{11} + A_{12})B_{22} + (A_{12} - A_{22})(B_{21} + B_{22})$$

$$A_{11}B_{11} + A_{12}B_{21} = A_{11}B_{11} + A_{12}B_{21}$$

Therefore C_{11} can be written:

$$C_{11} = (A_{11} + A_{22})(B_{11} + B_{22}) + A_{22}(B_{21} - B_{11}) - (A_{11} + A_{12})B_{22} + (A_{12} - A_{22})(B_{21} + B_{22})$$

2) $C_{12} = A_{11}B_{12} + A_{12}B_{22}$

$$A_{11}B_{12} + A_{1,2}B_{22} = A_{11}B_{12} - A_{11}B_{22} + A_{11}B_{22} + A_{12}B_{22}$$

$$A_{11}B_{12} + A_{1,2}B_{22} = A_{11}(B_{12} - B_{22}) + (A_{11} + A_{12})B_{22}$$

$$A_{11}B_{12} + A_{1,2}B_{22} = A_{11}B_{12} + A_{12}B_{22}$$

Therefore C_{12} can be written:

$$C_{12} = A_{11}(B_{12} - B_{22}) + (A_{11} + A_{12})B_{22}$$

3) $C_{21} = A_{21}B_{11} + A_{22}B_{21}$

$$A_{21}B_{11} + A_{22}B_{21} = A_{21}B_{11} + A_{22}B_{11} + A_{22}B_{21} - A_{22}B_{11}$$

$$A_{21}B_{11} + A_{22}B_{21} = (A_{21} + A_{22})B_{11} + A_{22}(B_{21} - B_{11})$$

$$A_{21}B_{11} + A_{22}B_{21} = A_{21}B_{11} + A_{22}B_{21}$$

Therefore C_{21} can be written:

$$C_{21} = (A_{21} + A_{22})B_{11} + A_{22}(B_{21} - B_{11})$$

4) $C_{22} = A_{21}B_{12} + A_{22}B_{22}$

$$A_{21}B_{12} + A_{22}B_{22} = A_{11}B_{11} + A_{11}B_{22} + A_{22}B_{11} + A_{22}B_{22} - A_{21}B_{11} - A_{22}B_{11} + A_{11}B_{12} - A_{11}B_{22} + A_{21}B_{11} + A_{21}B_{12} - A_{11}B_{11} - A_{11}B_{12}$$

$$A_{21}B_{12} + A_{22}B_{22} = (A_{11} + A_{22})(B_{11} + B_{22}) - (A_{21} + A_{22})B_{11} + A_{11}(B_{12} - B_{22}) + (A_{21} - A_{11})(B_{11} + B_{12})$$

$$A_{21}B_{12} + A_{22}B_{22} = A_{21}B_{12} + A_{22}B_{22}$$

Therefore C_{22} can be written:

$$C_{22} = (A_{11} + A_{22})(B_{11} + B_{22}) - (A_{21} + A_{22})B_{11} + A_{11}(B_{12} - B_{22}) + (A_{21} - A_{11})(B_{11} + B_{12})$$

After the result matrix C is transformed, consider the result of the transformation. There are 7 types of multiplication from the transformation results. Where :

1) $C_{11} = (A_{11} + A_{22})(B_{11} + B_{22}) + A_{22}(B_{21} - B_{11}) - (A_{11} + A_{12})B_{22} + (A_{12} - A_{22})(B_{21} + B_{22})$
2) $C_{12} = A_{11}(B_{12} - B_{22}) + (A_{11} + A_{12})B_{22}$
3) $C_{21} = (A_{21} + A_{22})B_{11} + A_{22}(B_{21} - B_{11})$
4) $C_{22} = (A_{11} + A_{22})(B_{11} + B_{22}) - (A_{21} + A_{22})B_{11} + A_{11}(B_{12} - B_{22}) + (A_{21} - A_{11})(B_{11} + B_{12})$

This is where $M_1 - M_7$ is formed. From the 7 multiplication combinations above, M_1 to M_7 will be determined based on the column. As a result, we get:

$M_1 = (A_{11} + A_{22})(B_{11} + B_{22})$

$M_2 = (A_{21} + A_{22})B_{11}$

$M_3 = A_{11}(B_{12} - B_{22})$

$M_4 = A_{22}(B_{21} - B_{11})$

$M_5 = (A_{11} + A_{12})B_{22}$

$M_6 = (A_{21} - A_{11})(B_{11} + B_{12})$

$M_7 = (A_{12} - A_{22})(B_{21} + B_{22})$

So that the result matrix C using the Strassen algorithm is obtained:

1) $C_{11} = (A_{11} + A_{22})(B_{11} + B_{22}) + A_{22}(B_{21} - B_{11}) - (A_{11} + A_{12})B_{22} + (A_{12} - A_{22})(B_{21} + B_{22})$
   $C_{11} = M_1 + M_4 - M_5 + M_7$
2) $C_{12} = A_{11}(B_{12} - B_{22}) + (A_{11} + A_{12})B_{22}$
   $C_{12} = M_3 + M_5$
3) $C_{21} = (A_{21} + A_{22})B_{11} + A_{22}(B_{21} - B_{11})$
   $C_{21} = M_2 + M_4$
4) $C_{22} = (A_{11} + A_{22})(B_{11} + B_{22}) - (A_{21} + A_{22})B_{11} + A_{11}(B_{12} - B_{22}) + (A_{21} - A_{11})(B_{11} + B_{12})$
   $C_{22} = M_1 - M_2 + M_3 + M_6$

So we already know how Strassen's algorithm is formed. Next, the Strassen algorithm will be applied to square matrix multiplication.

3.2 Application of Strassen's Algorithm to Square Matrix Multiplication

The application of the Strassen algorithm to the process of multiplying a square matrix of order 2^n. The steps in performing the square matrix multiplication process using the Strassen algorithm are:

1) The divide concept is used to divide the order of each square matrix A and square matrix B into $\left(\frac{n}{2}\right) \times \left(\frac{n}{2}\right)$.
2) Using the conquer concept in performing the matrix addition process by calculating M_i
   a. $M_1 = (A_{1,1} + A_{2,2}) \cdot (B_{1,1} + B_{2,2})$
   b. $M_2 = (A_{2,1} + A_{2,2}) \cdot B_{1,1}$
   c. $M_3 = A_{1,1} \cdot (B_{1,2} - B_{2,2})$
   d. $M_4 = A_{2,2} \cdot (B_{2,1} - B_{1,1})$
   e. $M_5 = (A_{1,1} + A_{1,2}) \cdot B_{2,2}$
   f. $M_6 = \left(A_{2,1} - A_{1,1}\right) \cdot \left(B_{1,1} + B_{1,2}\right)$
   g. $M_7 = (A_{1,2} - A_{2,2}) \cdot (B_{2,1} + B_{2,2})$
3) Calculate the result matrix C by substituting the result of the summation of M_i

$$C_{1,1} = M_1 + M_4 - M_5 + M_7; \qquad C_{1,2} = M_3 + M_5$$

$$C_{2,1} = M_2 + M_4; \qquad C_{2,2} = M_1 - M_2 + M_3 + M_6$$

4) Compile the result of C_i into the result matrix C to produce a complete solution.

$$\begin{bmatrix} C_{1,1} & C_{1,2} \\ C_{2,1} & C_{2,2} \end{bmatrix}$$

Furthermore, computational assistance will be used in the form of web-based applications using the javascript language. The programming language is used to see if the algorithm used can work efficiently. Using computational assistance can make it easier and faster for us to perform square matrix calculation operations, besides that with computational assistance we can validate the work we have done.

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\ 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 \\ 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32 \\ 33 & 34 & 35 & 36 & 37 & 38 & 39 & 40 \\ 41 & 42 & 43 & 44 & 45 & 46 & 47 & 48 \\ 49 & 50 & 51 & 52 & 53 & 54 & 55 & 56 \\ 57 & 58 & 59 & 60 & 61 & 62 & 63 & 64 \end{bmatrix}; \; B = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

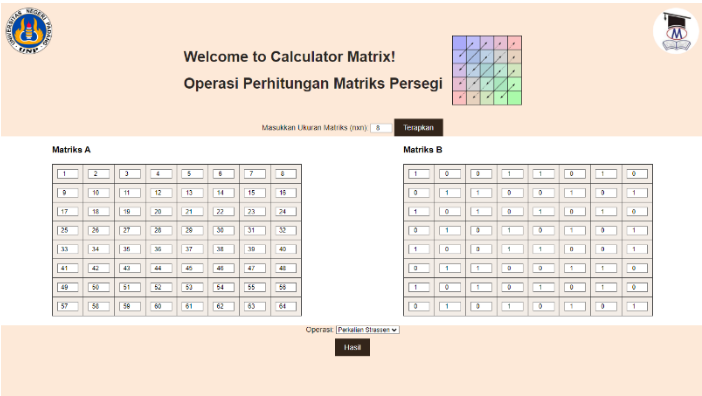

Fig 2. Web-based application appearance

Figure 1 is a display of a web-based application by entering the input elements of the 8×8 matrix. The C result matrix output that appears when the program is run is :

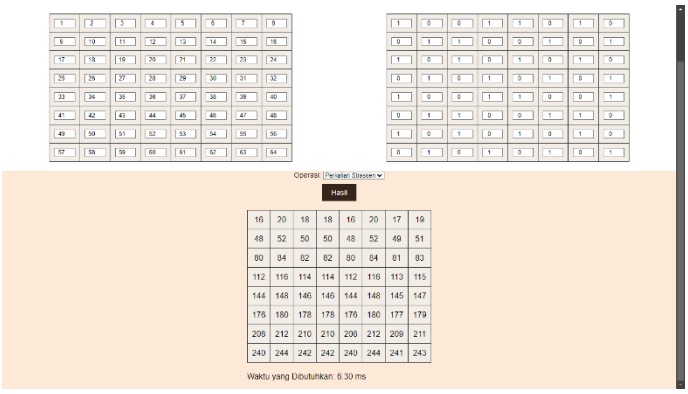

Fig 3. Multiplication Results Using Web-Based Applications

4 Conclusion

The identity transformation performed in the 2×2 matrix multiplication step is generally the origin of Strassen's algorithm. Strassen's algorithm is obtained from algebraic identity transformation by applying the properties that apply to algebra. Algebraic identity transformation is used to change the form of an equation without changing the original value. The transformation resulted in 7 multiplication processes denoted M_1 to M_7. Where the determination of M_i is guided by the column. Starting from column 1 row 1, column 1 row 2, and so on, with the condition that Mi cannot be the same.

The multiplication process step using the Strassen algorithm, which is summarized into 7 steps compared to the general one that requires 8 steps, can speed up the multiplication process, especially on matrices with large orders. So, the application of the Strassen Algorithm in multiplication, especially square matrix multiplication, can be an alternative to completing the square matrix multiplication process.

References

[1] Anton, Howard and Rorres, C. (1995). *Elementary Linear Algebra Applications Version, eight edition*. New York: John Willey & Sons, Inc

[2] Badan Pengembangan dan Pembinaan Bahasa. (2016). *Kamus Besar Bahasa Indonesia* (5th ed.). Accessed from https://kbbi.kemdikbud.go.id

[3] Robinson, John. 1996. *Structural Matrix Analysis for the Engineer*. John Willey & Sons : New York.

[4] Neapolitan, R. *Foundation of Algorithms, fifth edition*. Burlington: Jones & Bartlett Learning; 2015.

[5] Permatasari, D., & Harta, I. (2018). *Kemampuan berpikir aljabar siswa sekolah pendidikan dasar kelas V dan kelas VII: Cross-sectional study*. Jurnal Pendidikan dan Kebudayaan, 3(1), 1–15.

[6] Antoniades, A. C. (1990). *Poetics of Architecture: Theory of Design.* New York: Van Nostrand Reinhold.

[7] Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms (3rd ed.). The MIT Press.

[8] Zulfiansyah, M. B. P. (2023). Analisis Algoritma Divide and Conquer pada Permasalahan Penyusunan Data Secara Terurut. Institut Teknologi Bandung. Diakses dari https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2022-2023/Makalah/Makalah-Stima-2023-%2821%29.pdf

[9] Bressers, R. (2020). The *Strassen* Algorithm and Its Computational Complexity. *Journal BSc Mathematics & Application of Utrecht University*.

[10] Ikenmeyer, C., & Lysikov, V. (2019). Strassen's 2×2 matrix multiplication algorithm: A conceptual perspective. Annali dell'Università di Ferrara, 65, 241–248. https://doi.org/10.1007/s11565-019-00318-1

[11] D'Alberto, P. (2023, December 20). *Strassen*'s Matrix Multiplication Algorithm Is Still Faster. 1-8.

[12] Yanti, E. N., & Helmi. (2014). Perkalian Matriks Persegi Menggunakan Algoritma Strassen. *Buletin Ilmiah Mat. Stat. dan Terapannya (Bimaster), 03*, 201-208.

[13] Fajar Shodik, & Arnelis. (2021, September). *Analisis Algoritma Bahasa Pemrograman Java dengan Bahasa Pemrograman Visual Basic dan Program Aplikasi Perkalian Matriks*. Journal Of Mathematics UNP Vol. 6 No 3, 32-37.

[14] Joshua A., & Cristopher Moore. (2017, September 1). *Designing Strassen's Algorithm. Computer Science: Data Structures and Algorithms*, 1-10.

[15] Klappenecker, & Andreas. (n.d.). *Strassen's Matrix Multiplication Algorithm*. Retrieved November 11, 2024, from http://faculty.cs.tamu.edu/klappi/629/*Strassen*1.pdf

[16] Bernstein, S., & Bernstein, R. (1999). *Schaum's Outline of Theory and Problems of Elements of Statistics. I, Descriptive Statistics and Probability*. New York.

[17] John, R. (1996). *Structural Matrix Analysis for the Engineer*. New York: John Willey & Sons.

[18] Muhamad Fatwa, Rizki Ristu, Sriwinarty Pandiangan, & Edi Supriyadi. (2022, Desember). *Pengaplikasian Matlab Pada Perhitungan Matriks. Papanda Journal of Mathematics and Sciences Research* (PJMSR), 81-93.

[19] Munir, Rinaldi. (n.d.). Algoritma Divide and Conquer . Retrieved November 11, 2024, from http://kur2003.if.itb.ac.id/file/Kompleksitas%20Algoritma.doc

[20] Seymour Lipschutz, P., & Marc Lipson, P. (2009). Schaum's Outlines: Linear Algebra (4th ed.).