Sumatran *batik* motif design and documentation using turtle graphics algorithm based on local wisdom

Ratnadewi Ratnadewi [a,1,*], Agus Prijono [a,2], Ariesa Pandanwangi [a,3]

[a] Universitas Kristen Maranatha, Jl. Surya Sumantri 65, Bandung 40164, Indonesia
[1] ratnadewi@eng.maranatha.edu*; [2] agus.prijono@eng.maranatha.edu; [3] ariesa.pandanwangi@maranatha.edu
* corresponding author

ARTICLE INFO

Article history
Received 2024-09-15
Revised 2024-12-16
Accepted 2024-12-26

Keywords
Batik Sumatra
Digitalization
Motifs
Program
Turtle Graphics

ABSTRACT

Using the Turtle Graphic algorithm to digitize and develop Sumatra batik motifs is an innovative step in preserving local wisdom. This algorithm, which was originally used in graphics programming to educate children about computer concepts, is now being utilized to design and reproduce intricate and detailed *batik* motifs. The problem is that most existing *batik* motifs have not been digitally stored. Therefore, the purpose of the research here is to digitize Sumatra *batik* motifs using the Turtle Graphics algorithm. Turtle graphics utilizes command-based programming principles to draw geometric shapes. As an icon for North Sumatra Province, the basic motif of the existing water tower *batik* is the focus of this pattern accuracy. It is documented using a Python program that uses the turtle graphics algorithm, and a new Sumatran *batik* motif design is created from this basic motif. The benefits of research results with this approach are that designers can digitally modify and reproduce traditional batik motifs. This algorithm allows the drawing of patterns with high accuracy, making creating consistent and precise motifs easier. This digitization process also helps document and preserve *batik* motifs. The use of this technology not only speeds up the design process but also provides wider access to the younger generation and the international community to appreciate and learn about Sumatra *batik*. By integrating modern technology and local wisdom, the digitization of rare *batik* motifs has the potential to strengthen cultural identity and increase the economic value of traditional *batik* products. The specific steps in the motif digitization or pattern-making process are as follows: determine the motif to be digitized, select the starting point of the moving turtle, determine the turtle's direction of motion in relation to the curve to be formed, execute the program, and if the turtle curve does not move in the desired direction, change the program and execute again until the turtle motion curve matches the desired curve. This process is repeated multiple times.

1. Introduction

Transmigrants from Java to Sumatra, some of whom were *batik* artisans, are credited with starting the tradition of Sumatran *batik*. They created *batik* motifs on the island of Sumatra, drawing inspiration from the local landscape. As a result, *batik* is now a craft on the island of Sumatra. The Batak and Nias tribes' handicrafts can be seen in items such as clothes, battle gear, and carvings seen in homes. *Batik* artisans use this design to create drawings on *batik* fabric. Although the design of this motif is less developed, this article explores how to store it as a Python program using the turtle graphics method. This allows the motif to be stored with minimal storage memory while maintaining a high level of quality in vector images. Compared to manual design, the design creation process is facilitated digitally by having the fundamental batik motifs. This motif can be transformed into a modern batik design without sacrificing the historical concept.

Using the turtle graphic algorithm to digitize and develop Sumatra batik motifs is an innovative step in preserving local wisdom. To be able to create a motif with the turtle graphics algorithm, someone needs to learn a programming language, one of which is Python. This programming needs to be done so that when the program is executed, the result of the execution is a *batik* motif image. A programmer must have logical analysis, language, visual, and creativity skills. This is related to the abilities of the left and right brain. This algorithm, originally used in graphics programming to educate children about computer concepts, is now utilized to design and reproduce complex and detailed *batik* motifs. Educating children to create *batik* motifs with the turtle graphics algorithm can train both the left and right brains.

The human brain consists of two main hemispheres: the left and right hemispheres. The left brain's main functions are often associated with language processing, logic, and analysis. It plays a role in speaking, reading, and writing, as well as mathematical and analytical skills. The characteristics of the left hemisphere are usually more involved in activities that require structured and systematic thinking. This includes the ability to follow instructions linearly and structure information in an organized way [1], [2]. Programming activities involve logical, systematic, and analytical thinking skills, which are the dominant functions of the left brain. This includes writing code, debugging, and solving problems that require structured thinking. The left hemisphere manages the mathematical and logical operations that are essential in software and algorithm development [3].

The divide of cognitive and motor processes between the left and right hemispheres of the brain, each of which plays a distinct role, is known as brain lateralization. Brain lateralization illustrates how specific cognitive processes are more prevalent in particular hemispheres in this circumstance. The cognitive processes of language processing, logical analysis, and methodical problem-solving are more prevalent in the left hemisphere. Programming-related cognitive functions: The left hemisphere's predominance significantly impacts the capacity to comprehend the syntax of programming languages, the logic of algorithms, and the structure of code. This hemisphere is also linked to executive activities, including organization and planning [4]. Because they can reason clearly and sequentially, those with left-brain dominance are more likely to learn programming languages, according to research [5].

On the other hand, the cognitive processes of creativity and visual processing are more prevalent in the right hemisphere—cognitive processes related to creativity, imagery, spatial processing, and holistic thinking. The right hemisphere is more involved in areas of software development or user interface (UI/UX)-based design and programming that demand visual imagination and creativity. In contrast, programming is typically linked with logic, which is more affected by the left hemisphere. This demonstrates how crucial innovation is to creating effective and approachable software solutions. According to research, programmers who create apps with creative layouts or user-friendly interfaces typically employ both hemispheres in balance [6]. The most effective programmers typically strike a decent balance between both hemispheres. The capacity to think methodically (left brain) and creatively (right brain) enables them to solve more complicated programming issues and generate more efficient and novel solutions.

Meanwhile, the main functions of the right hemisphere are more often associated with creativity, intuition, and visual-spatial information processing. It plays a role in understanding patterns, music, art, and design and in handling tasks that require holistic and nonverbal thinking. The right brain is often associated with the ability to perform visual and spatial processing and creativity, such as understanding shape, color, and layout, which are essential elements in the visual arts [7]. Right hemisphere characteristics tend to focus more on the emotional and contextual aspects of experience, which is essential in art and performance. Connecting seemingly unrelated ideas and seeing the big picture is vital [8], [9]. The presentation confirms that the digitization process in making *batik* motifs using Turtle Graphics is closely related to the main functions of the right brain hemisphere.

The right brain is known to play a role in creativity, intuition, and visual-spatial processing abilities that are very important in the arts, including in digitally designing *batik* motifs. Turtle Graphics allows designers to create *batik* patterns by utilizing programming code that generates a gradual visualization of the pattern. The strong right-brain function for recognizing shape, color, and layout assists the designer in creating themes with aesthetic value and a balanced composition. The technique also incorporates holistic thinking, which allows designers to combine traditional *batik* features with new modern ways. The arrangement of motif layouts relies heavily on the right-brain's visual and spatial creative process, which ensures that each part is well connected. The right brain's ability to capture emotional and contextual components assists designers in understanding the philosophical meaning underlying each pattern, resulting in patterns that are both visually appealing and culturally significant. The digitization of *batik* by Turtle Graphics promotes more creative development. This approach not only improves visual abilities and intuition, but it also creates new potential to showcase *batik* as a valuable artwork in the digital age. This is supported by the findings of research [10].

While there is a tendency for the left and right hemispheres to function differently, it is important to note that they do not work in isolation. Complex cognitive functions require collaboration between both hemispheres, and this connectedness is important for the overall balance of brain function [11]. Many activities, such as art and programming, require integration between both hemispheres. For example, software design requires creativity (right brain) to design an attractive user interface and logical functioning (left brain) to ensure that the design works properly. Digital art creation requires an understanding of visual design principles (right brain) as well as technical skills in using software (left brain) [12]. The right and left brains play complementary roles in various activities, be it art or programming. Collaboration between the fields of art and engineering in the digital creation of *batik* motifs provides many benefits, especially in terms of creativity, efficiency, and innovation [13]. The main benefits of this collaboration are.

- By combining artists' creativity and digital designers' technical expertise, the resulting *batik* motifs can become more innovative and interesting. Artists can explore creative and aesthetic ideas, while digital techniques allow the implementation of these ideas in a more varied and detailed manner.
- Digital techniques allow artists to explore a wide variety of designs and patterns quickly and easily. For example, artists can use design software to experiment with colors, textures, and compositions that may be difficult to do manually [14]. Digital techniques enable creating and modifying *batik* motifs with speed and high precision. Designs can be developed, tested, and customized in a shorter time compared to traditional methods. With digital design, *batik* motifs can be reproduced consistently without loss of quality. This is important for mass production, where uniformity of design is highly desired [15].
- Digital techniques make it possible to digitize traditional *batik* motifs, which helps preserve and disseminate cultural heritage. The motifs can be stored in digital format for archival purposes and future reference. This collaboration also opens up opportunities to combine traditional *batik* designs with modern design elements, resulting in works that are relevant to current trends without ignoring cultural roots [16].
- Through digital techniques, batik motifs can be easily accessed and shared through online platforms, expanding market reach and easing distribution. Digital technology enables collaboration between designers and artists from different parts of the world, facilitating the exchange of ideas and techniques that enrich batik designs [17].
- Digital techniques allow the creation of customized *batik* designs to meet customers' specific needs. This can include modifications to colors, motifs, and other design elements according to individual preferences. With design software, artists can easily create prototypes and visualizations of motifs before production, allowing for quick fixes and adjustments [18].

Overall, the collaboration between art and engineering in the digital creation of *batik* motifs results in more innovative, efficient, and relevant designs and supports the preservation and dissemination of cultural heritage while utilizing the power of modern technology.

2. Method

The method of creating digital *batik* motifs using the Turtle Graphics algorithm is an approach that combines design art with programming techniques. Turtle Graphics is a drawing method often used in teaching computer programming, where a ‘turtle’ (frog) moves on the screen to draw patterns based on given commands. Turtle Graphics can be used to create motifs in a structured and automated way in the context of digital batik. The steps of making digital *batik* motifs with Turtle Graphics: (1). Introduction to Turtle Graphics. Turtle Graphics is a drawing method that allows the creation of images by controlling a ‘turtle’ that moves around the screen. Commands such as forward, backward, rotate right, rotate left, raise the pen, and lower the pen direct the turtle to draw various shapes and patterns. Turtle Graphics is often used in the Python programming language, but the principles can be applied in a variety of programming languages with appropriate graphics libraries [19], [20]; (2). Decide on the *batik* motif to be drawn. *Batik* motifs often have repetitive and symmetrical patterns, such as floral, geometric, or abstract. *Batik* often involves complex geometry elements. Determine how the basic shapes can be combined or repeated to create the desired pattern [21], [22]; (3). Implementation with Turtle Graphics needs to prepare programming with the Turtle Graphics library in Python. Use the Turtle Graphics command to draw *batik* patterns. Adjust parameters such as size, shape, and angle to create a wide variety of *batik* patterns. Use the Turtle Graphics function to change the curves on the lines forming the motif [23]; (4). Once the *batik* motif is complete, the program can be saved, and the resulting image can be processed using an image processing application. The turtle library in Python for drawing graphs can be done by giving commands forward, backward, rotate left or right how many degrees, lower or raise the pen, go to position (*x*,*y*), and iteration (Fig. 1). Explanation about parameter values, algorithmic logic, or special coding functions can be seen in detail at Table 1 about our research methods. For complex patterns, mathematical geometry models are used, for example, to determine whether the turtle turns right or left by a certain degree, how many steps the turtle needs to move, the turtle's starting position, and how many steps the turtle needs to return. Symmetry and repetition programs use looping to allow the same procedure to be repeated multiple times. Functions can be used to separate every part, allowing the program to be called repeatedly in the main program. The assessment criteria and testing processes are described by comparing program execution outcomes to the original *batik* motif images, whilst the feedback mechanism from *batik* specialists to strengthen the methodological framework is gained from the motifs generated using turtle graphics.

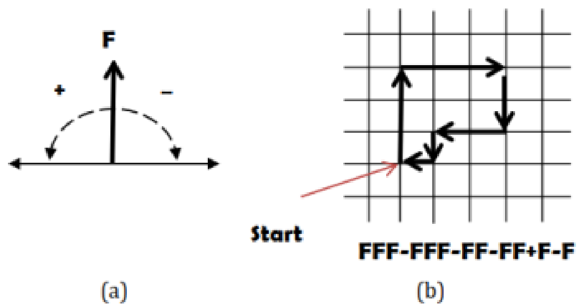


Fig. 1. (a) Interpretation of turtle graphics F for forward step, + for left direction angle, and - for right direction angle, (b) Writing the direction of motion with - for angle 90°*)*

Some simple turtle commands [19], [24] are as follows: (1) Forward (5) = F(5) = FFFFF is a turtle motion forward by 5 pixels; (2) Backward (4) = B(4) = BBBB is a backward turtle motion by 4 pixels; (3) Right (20) = -D(20) is the motion of the turtle rotating clockwise by 20 degrees; (4) Left (40) = +D(40) is the motion of the turtle rotating counterclockwise by 40 degrees; (5) Goto (x,y) to move the turtle to position x, y; (6) Penup () = Up is to disable the pen so that when there is movement of the turtle, it does not create a graph; (7) Pendown () = Down is to activate the pen so that when the turtle is moved, it will create a graph; (8) Setheading(20) = Set the turtle head at an angle of 20 degrees; (9) a : FFF -D(6) is looping a program containing forward three times and turning right 6 degrees, the letter a can be replaced with other letters and can be a combination of letters and numbers, to distinguish one looping from another. Before you begin to format your paper, first write and save the content as a separate text file. Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit the use of hard returns to only one return at the end of a paragraph. Do not add any pagination anywhere in the paper. Do not number text heads- the template will do that for you. The method of creating digital *batik* motifs with Turtle Graphics utilizes programming principles to produce aesthetic and structured designs. Using command graphics to draw batik motifs, valuable historical motifs can be created efficiently. This enables broader and more innovative design exploration and speeds up the process of creating digital *batik* motifs.

3. Results and Discussion

The motif that will be discussed here is the *Tirtanadi* water tower motif. The *Tirtanadi* Water Tower is a historical building in Medan, North Sumatra, Indonesia. The water tower was built in 1908 by the Dutch government as part of the water storage and distribution system in Medan City. The *Tirtanadi* Water Tower building has a distinctive architecture and has become a tourist attraction as well as a local historical relic. The motif of the *Tirtanadi* Water Tower in Malay *batik* symbolizes the pride of the people of Medan in the existence of the local iconic building. This is a form of recognition of the importance of the *Tirtanadi* Water Tower as one of the important buildings in Medan city. The water tower theme is a captivating subject for interdisciplinary analysis. Culturally, it embodies both industrial heritage and communal resilience. Artistically, it combines form and function, offering a space for artistic expression while being grounded in practical necessity. Modern water tower design relies significantly on advanced computing technologies to improve structural integrity, fluid dynamics, and sustainability. Water towers' cultural significance has grown from utilitarian items to symbols of development and identity, and their artistic value stems from their ability to combine engineering perfection with aesthetic care. Finally, their current computational precision ensures they continue acting as efficient, sustainable, and safe structures in our built surroundings. The water tower program can be seen in Fig. 2.

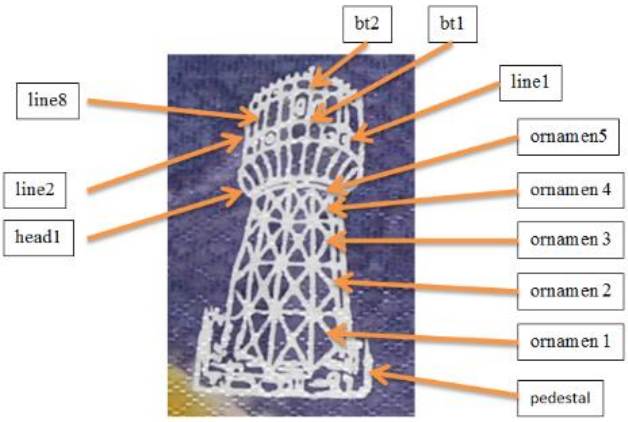
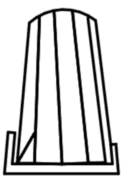


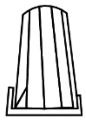
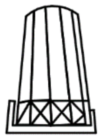
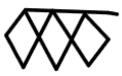
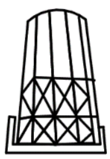
Fig. 2. Functions used in the turtle graphics program [25]

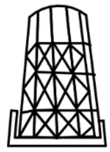
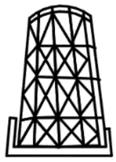
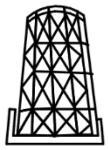
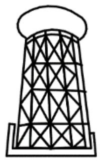
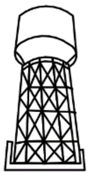
The result of executing the program in Table 1 can be seen in Fig. 3.

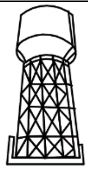
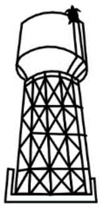
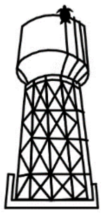
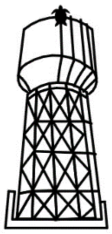
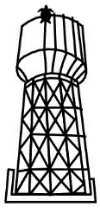
Table 1. Turtle graphics algorithm on the water tower motif

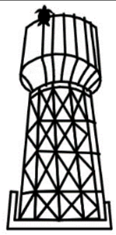
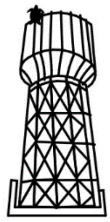
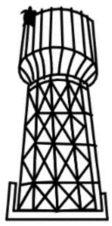
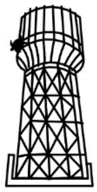
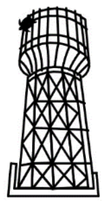
Function Name and kurva *yang dibentuk*	Turtle Graphics Algorithm	Iteration	Description
Pedestal 	Goto(500,400) Setheading(85) Down() F(40) –D(95) F(165) –D(95) F(60) –D(90) F(10) –D(90) F(45) +D(95) F(145) +D(95) F(28) –D(90) F(8) Up() B(8) +D(92) Down() F(180) +D(50) a(8) +D(38) F(203) +D(155) F(50) B(50) –D(155) +D(95) F(25) +D(88) F(212) B(212) –D(90) F(40) +D(95) F(219) B(219) –D(90) F(38) +D(93) F(219) B(219) –D(95) F(25) +D(96) F(210) Up()	a: F(13) +D(10)	Pedestal is the name of the function used to create the base of the water tower motif. F(5)=Forward (40), turtle motion forward by 40 pixels B(4)=Backward (4), backward turtle motion by 4 pixels -D(95)=Right (95) is the motion of the turtle rotating clockwise by 20 degrees.

Vol. 6, No. 2, December 2024, pp. 248-260

Function Name and kurva *yang dibentuk*	**Turtle Graphics Algorithm**	**Iteration**	**Description**
			+D(95)=Left (95) is the motion of the turtle rotating counterclockwise by 40 degrees. Goto (500,400) to move the turtle to position 500, 400. Penup () = Up is to disable the pen so that when there is movement of the turtle, it does not create a graph. Pendown () = Down is to activate the pen so that when the turtle is moved, it will create a graph. Setheading(85) The program execution results are as follows: 
ornamen1 	Goto(490,380) Setheading(180) Down() F(108) +D(10) F(20) B(20) +D(120) F(60) B(60) –D(128) B(38) +D(50) F(55) B(55) +D(80) F(62) +D(90) F(28) lt(92) F(35) +D(96) F(60) Up()		Making the bottom decoration (1) The program execution results are as follows: 
ornamen 2 	Goto(485,320) Setheading(190) Down() F(10) +D(8) F(90) +D(55) F(35) +D(65) F(40) +D(125) F(70) -D(120) F(72) +D(110) F(36) +D(80) F(38) +D(110) F(68) -D(125) F(72) Up()		Making the bottom decoration (2) The program execution results are as follows: 
ornamen 3 	Goto(478,270) Setheading(200) Down() F(10) +D(20) F(82) +D(50) F(28) +D(75) F(30) +D(115) F(56) -D(118) F(62) +D(115) F(34) +D(82) F(30) +D(100) F(54) -D(120) F(55) Up()		Making the bottom decoration (3) The program execution results are as follows:

Function Name and kurva *yang dibentuk*	**Turtle Graphics Algorithm**	**Iteration**	**Description**
			
ornamen 4 	Goto(470,228) Setheading(213) Down() a(8) +D(65) F(40) +D(136) F(58) -D(127) F(58) +D(127) F(47) Up()		Making the bottom decoration (4) The program execution results are as follows: 
ornamen 5 	Goto(478,270) Setheading(200) Down() F(15) -D(43) F(53) +D(123) F(53) -D(123) F(53) Up()		Making the bottom decoration (5) The program execution results are as follows: 
Head 	Goto(474,222) Setheading(-40) Down() c(7) -D(18) e(7) c(7) Up()	c: F(8) +D(18) e: F(15) +D(10)	Making the top of the tower head (1) The program execution results are as follows: 
kepala2 	Goto(465,218) Setheading(-45) Down() F(30) +D(50) F(80) +D(50) e(8) +D(38) F(75) +D(55) g(4) Up()	g: F(8) -D(8)	Making the top of the tower head (1) The program execution results are as follows: 
line1 	Goto(459,214) Setheading(-60) Down() h(4) F(3) +D(53) F(78) Up()	h: F(8) -D(4)	Making support poles (1) The program execution results are as follows:

Function Name and kurva *yang dibentuk*	**Turtle Graphics Algorithm**	**Iteration**	**Description**
line2 	Goto(450,209) Setheading(-65) Down() i(4) F(3) +D(42) F(78) Up()	i: F(8) +D(3)	 Making support poles (2) The program execution results are as follows:
line3 	Goto(438,206) Setheading(-78) Down() j(4) F(3) +D(28) F(73) Up()	j: F(9) -D(3)	 Making support poles (3) The program execution results are as follows:
line4 	Goto(424,203) Setheading(-85) Down() k(4) F(3) +D(15) F(66) Up()	k: F(10) -D(2)	 Making support poles (4) The program execution results are as follows:
line5	Goto(412,205) Setheading(-90) Down() l(4) F(3) -D(10) F(67) Up()	l: F(10) +D(2)	 Making support poles (5) The program execution results are as follows:
line6	Goto(400,210) Setheading(-90) Down() m(4) F(3) -D(22) F(70) Up()	m: F(10) +D(5)	 Making support poles (6) The program execution results are as follows:

Function Name and kurva *yang dibentuk*	**Turtle Graphics Algorithm**	**Iteration**	**Description**
			
line7 	Goto(390,210) Setheading(-92) Down() n(4) F(3) -D(29) F(68) Up()	n: F(10) +D(6)	Making support poles (7) The program execution results are as follows: 
line8 	Goto(382,212) Setheading(-94) Down() o(4) F(3) -D(45) F(67) Up()	o: F(9) +D(9)	Making support poles (8) The program execution results are as follows: 
bt1 	Goto(482,170) Setheading(-140) Down() p(9) Up()	p: F(15) +D(10)	Making the upper bone The program execution results are as follows: 
bt2 	Goto(482,135) Setheading(-140) Down() p(8) Up()		Making the upper bone The program execution results are as follows: 

The result of executing the program in Table 1 can be seen in Fig. 3. After the motif is created, it can be used for *batik* design. The composition and patterns in *batik* are arranged in a repetitive manner. The water tower is arranged in the same pattern and repeats, creating a rhythmic impression and balance in this *batik* design.

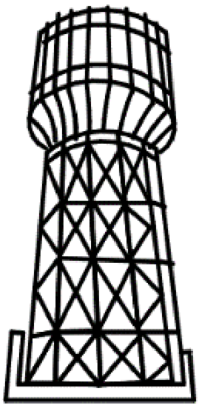


Fig. 3. Results of program execution with turtle graphics algorithm.

In Fig. 4, it can be seen that the water tower pattern is made to vary with the vine motif placed between the water tower *batik* patterns so that it appears to be alternating in size and orientation. The slope in the arrangement of objects gives a dynamic and interesting visual effect. It provides variations in the visual objects of *batik* motifs so that they are consistent with the themes carried in this *batik* motif. Green shades with yellow variations give a cool and dynamic impression. The green color seems to dominate this *batik*, giving a natural and fresh impression. The green color is also often associated with life, coolness, and balance, so psychologically, it will provide comfort to the wearer [26]. Color schemes can be created for background and line colors, but only the basic motif is stored; for designs with vast picture fields, the motif is copied to a commonly used vector editing application, as the motif implemented by the Python program is a vector image type.

Fig. 4. The result of the *batik* design of the water tower slope (*lereng menara air*) motif [27]

Digital *batik* motifs created using the Turtle Graphics algorithm offer various benefits in terms of design, production, and education. Here are some of the main benefits of using the Turtle Graphics algorithm in creating digital *batik* motifs: (1). Efficiency in Design: the Turtle Graphics algorithm makes it easy to create repetitive patterns that are common in *batik*. Using loops and graphic commands, a consistent pattern can be quickly generated without manually drawing each element. Designers can easily change parameters such as size, angle, and color to experiment with different design variations. This allows for faster and more efficient design exploration than traditional methods; (2) Turtle Graphics is an excellent tool for teaching basic programming and math concepts. It helps beginners understand how algorithms work and how geometric designs can be generated through code. The use of Turtle Graphics in art and design education allows students to learn *batik* design principles in an interactive and technology-based way; (3) Turtle Graphics allows designers to create complex and innovative patterns by combining design elements that are impossible to achieve by hand. This opens up opportunities to create unique and interesting *batik* motifs. By applying algorithms in *batik* design, you can explore new design ideas that you may not have thought of before. This technique encourages creative exploration through programming; (4) Digitalization of Cultural

Heritage can be done using digital techniques such as Turtle Graphics, and traditional *batik* motifs can be digitized and preserved more easily. This also makes disseminating and promoting batik motifs to a global audience easier. Digital methods allow the adaptation of traditional *batik* motifs to reflect modern design trends without abandoning the motif's distinctive features so that *batik* remains relevant and evolves with the times; (5) Customization and Personalization can be done with algorithms; designers can easily create customized *batik* designs according to customer needs or preferences. This allows the creation of unique and specific motifs. Design digitization enables rapid prototyping and design modification without having to recreate from scratch. This increases flexibility in the design and production process.

According to the explanation above, the digital method might assist in preserving and documenting *batik* motifs, yet this documentation technique may reduce the authenticity of traditional *batik*-producing practices. In this scenario, turtle graphic-based digitization techniques, up to storage in digital databases, can provide an excellent opportunity to create *batik* motifs rich in cultural values that can be precisely documented, made worldwide accessible, and protected from physical damage. Academics are concerned about how digitization would affect the authenticity of traditional *batik* creation. Traditional techniques, such as *batik* making using canting or stamping, need manual skills and detailed processes that reflect philosophical ideals and locale. Although digitization is efficient, it has the potential to eliminate uniqueness since digital patterns can be replicated in large numbers without requiring direct human participation. This increases the possibility of cultural meanings being lost in traditional behaviors. Academics are concerned about how digitization would affect the authenticity of traditional *batik* creation. Traditional techniques, such as *batik* making using canting or stamping, need manual skills and detailed processes that reflect philosophical ideals and locale. Although digitization is efficient, it has the potential to eliminate uniqueness since digital patterns can be replicated in large numbers without requiring direct human participation. This increases the possibility of cultural meanings being lost in traditional behaviors. Industrialization and globalization increasingly challenge cultural heritage preservation, but digital technology provides novel alternatives. Massive digitization programs, such as UNESCO's Digital Preservation of Historical Manuscripts, use digital tools to transform historical manuscripts and physical artifacts into digital formats, allowing for more access while minimizing the danger of destruction to the original objects. An article maintains the digital form of digital artifacts created through digitization, which is critical for worldwide access and long-term preservation [28], [29].

4. Conclusion

A significant finding in this study is that the basic theme of the water tower may be created using a Python program that takes a minimum storage capacity, has good image quality, and is in the form of a vector image. This theme can be utilized in different *batik* design editing software, allowing designers to generate new designs more easily. Using the Turtle Graphics algorithm in the creation of digital *batik* motifs provides a range of significant benefits, from efficiency and innovation in design to educational applications and cultural preservation. This technique enables the creation of complex motifs in a structured and creative way and expands the possibilities in *batik* design in terms of both aesthetics and functionality. The use of digital *batik* motifs created with the Turtle Graphics algorithm combines art and engineering approaches, creating benefits that include efficiency, innovation, and education. The following conclusions relate to how this method interacts with the right-brain and left-brain functions, with the right brain involving the creativity and design aspects of *batik* motifs. The Turtle Graphics algorithm enables the exploration and visualization of aesthetically complex *batik* patterns, enriching the experience of art and design innovation. Right-brain functions play a role in creating artistic and imaginative patterns and in interesting visual design experiments. Meanwhile, the left brain involves the technical and logical aspects of the Turtle Graphics algorithm. The programming process, the use of graphic commands, and the manipulation of design parameters require analytical and systematic abilities, which are functions of the left brain. Algorithms enable the application of mathematical and geometric principles to create precise and consistent designs.

With the Turtle Graphics algorithm, designers can easily integrate creative ideas (right brain) into digital *batik* designs while utilizing programming logic (left brain) to produce structured and repetitive patterns. This results in innovative and attractive *batik* designs. Digital techniques increase efficiency in *batik* motif-making with automation and design modification (left brain). Meanwhile, creativity in design exploration and pattern customization activates right-brain functions. This enables the creation of precise and consistent *batik* motifs without sacrificing aesthetic elements. The use of Turtle Graphics in education teaches programming and design skills (left brain) while facilitating the understanding of visual design principles and creativity (right brain). It also supports the preservation of *batik* cultural heritage through digitization, enabling wider access and distribution. Algorithms enable the creation of customized *batik* designs according to individual preferences, integrating creativity (right brain) with programming techniques (left brain) to produce unique and on-demand motifs. Digital processes enable the rapid creation and modification of prototypes, combining creative ideas (right brain) with analytical capabilities (left brain) for design customization and optimization. The use of digital *batik* motifs created with the Turtle Graphics algorithm reflects the collaboration between right and left-brain functions. It combines creativity and aesthetics with logical and systematic programming, resulting in innovative, efficient, and precise *batik* designs. This integration between art and engineering enriches the design process and supports education, cultural preservation, and design customization, creating broad and integrated benefits.

This research still has a number of drawbacks that researchers need to address, one of which is the restricted investigation of more sophisticated algorithm modifications, which limits its ability to produce more inventive and diverse designs. Furthermore, this study did not investigate in depth how this algorithm may be applied on an industrial scale or how its use can be accepted by the larger society, particularly traditional *batik* artisans because its deployment necessitates socialization and training for *batik* artisans. Critically, there has been no assessment of the impact of adopting digital technology in batik production, nor have there been constraints in its accessibility among certain communities, such as *batik* craftsmen. As a result, future research should examine merging the Turtle Graphics algorithm with user-friendly technology to increase accessibility for a larger community. In the future, this research can serve as a foundation for additional research into the prospect of collaboration between traditional *batik* craftspeople and digital designers to create works that mix traditional uniqueness and current innovation. The Turtle Graphics algorithm has the potential to become a key pillar in the digital advancement of *batik* design while also promoting cultural preservation through sustainable innovation. This study demonstrates that the programming ability to generate *batik* motifs using the turtle graphics algorithm involves both logic (controlled by the left brain) and creativity (controlled by the right brain), emphasizing the importance of cross-hemispheric collaboration in solving the task of creating *batik* motifs.

Acknowledgment

The research team would like to thank all the institutions funding this research and Universitas Kristen Maranatha Bandung for the support provided.

Declarations

Author contribution	**:**	All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.
Funding statement	**:**	This research fund comes from the Ministry of Education, Culture, Research and Technology for Fiscal Year 2024, 106/E5/PG.02.00.PL/2024 with No. Derivative Contract: 007/SP2H/RT-MONO/LL4/2024, 344-I/LPPM/UKM/VI/2024 dated 18 June 2024.
Conflict of interest	**:**	The authors declare no conflict of interest.
Additional information	**:**	No additional information is available for this paper.

References

[1] M. S. Gazzaniga, *The split brain revisited.* 2000. doi: 10.1038/scientificamerican0798-50, doi : 10.1038/scientificamerican0798-50.

[2] S. Pinker, “The Language Instinct.” Dutton, 2007.

[3] S. Pinker, *How the mind works*. London: Penguin Books, 1997.

[4] A. R. Luria, *The Working Brain : An Introduction To Neuropsychology*. Basic Books, 1973.

[5] R. E. Jung and R. J. Haier, “The Parieto-Frontal Integration Theory (P-FIT) of intelligence: Converging neuroimaging evidence,” *Behav. Brain Sci.*, vol. 30, no. 2, pp. 135–154, 2007, doi: 10.1017/S0140525X07001185.

[6] S. Beilock, *Choke: What the Secrets of the Brain Reveal About Getting It Right When You Have To*. 2010. doi: 10.1080/21520704.2011.564535.

[7] V. S. Ramachandran and S. Blakeslee, *Phantoms in the Brain, Probing the Mysteries of the Human Mind*. 1998.

[8] H. Gardner, *Frames of Mind: The Theory of Multiple Intelligences*. Basic Books, 1993.

[9] R. W. Sperry, “Cerebral organization and behavior,” *Science*, vol. 133, no. 3466. pp. 1749–1757, 1961. doi: 10.1126/science.133.3466.1749.

[10] M. Chirimuuta, *The Brain Abstracted*, no. December. 2024. doi: 10.7551/mitpress/13804.001.0001.

[11] M. S. Gazzaniga, “The Ethical Brain,” *J. R. Soc. Med.*, vol. 98, no. 9, pp. 433–434, 2005, doi: 10.1258/jrsm.98.9.433.

[12] D. J. Levitin, *This is your brain on music*, First edit. 2006.

[13] R. Ratnadewi, A. Pandanwangi, and A. Prijono, “Learning mathematics through art in a faculty of engineering,” *World Trans. Eng. Technol. Educ.*, vol. 19, no. 3, pp. 271–275, 2021.

[14] 3DTotal Publishing, *Digital Art Masters*, no. Volume 8. 3DTotal Publishing, 2013.

[15] A. S. M. Sayem, *Digital Fashion Innovations : Advances in Design , Simulation , and Industry*, no. April. Textile Institute Professional Publications Series, 2023. doi: 10.1201/9781003264958.

[16] J. McCarthy and P. Wright, *Technology as experience*, vol. 11, no. 5. 2004. doi: 10.1145/1015530.1015549.

[17] C. Paul, “Review of Christiane Paul ’ s 2003 ’ Digital Art ’ Thames & Hudson,” *Digit. Art*, vol. 224, no. January 2004, 2003.

[18] H. Yeung, T. Choi, and C. Chiu, *Innovative Quick Response Programs in Logistics and Supply Chain Management*, no. July. 2010. doi: 10.1007/978-3-642-04313-0.

[19] P. Elance, “Turtle graphics using Python,” *Tutorial points*, 2019.

[20] J. Zelle, *Python programming: an introduction to computer science*. Franklin, Beedle & Associates Inc, 2010.

[21] W. Wibawanto, T. Triyanto, A. Cahyono, and T. R. Rohendi, “Teaching the Creation of Batik Motifs to Foreign Students through Distance Learning,” *Int. J. High. Educ.*, vol. 10, no. 7, p. 1, 2021, doi: 10.5430/ijhe.v10n7p1.

[22] R. Dinata and Z. Fan, “Elaboration of Batik Pattern Design Application in Indonesia,” *Int. Humanit. Appl. Sci. J.*, vol. 2, no. 3, pp. 50–57, 2019, doi: 10.22441/ihasj.2019.v2i2.06.

[23] R. Goldman, S. Schaefer, and T. Ju, “Turtle Geometry in Computer Graphics and Computer Aided Design,” *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2013.

[24] R. Ratnadewi, A. Pandanwangi, and A. Prijono, *Implementasi bahasa Python pada motif batik Tasikmalaya*, Cetakan Pe. Bandung: Penerbit Yayasan Lembaga Gumun Indonesia viii, 2021.

[25] F. Nisa, “Batik Sumatera Utara , 20 + Motif Khas Batak dan Melayu beserta Maknanya,” 2023.

[26] G.A.Agoston, *Color Theory and Its Application in Art and Design*, vol. 11, no. 1. Springer, 1970.

[27] R. Ratnadewi, “Batik Sumatera Utara Motif Lereng Menara Air,” EC00202476802, 2024

[28] S. Cele, M. Stewart, and R. Asiedu, “Digital Archival Preservation and Cultural Heritage,” *J. Digit. Humanit. Assoc. South. Africa*, vol. 5, no. 1, 2024, doi: 10.55492/dhasa.v5i1.5017.

[29] J.-A. Beraldin *et al.*, “Virtual Reconstruction of Heritage Sites: Opportunities and Challenges Created by 3D Technologies,” *Int. Work. Rec. Model. Vis. Cult. Heritage.*, no. January, 2005.