Digital Double Exposure on Binary Images Analysis by Boolean Algebra Operations

Fitria Budi Ananda – 13517133
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13517133@itb.ac.id

***Abstract*—Blending two images into one single image produces an image known as double exposure image. In digital photography, this technique can be done by exposing a layer over another layer. For each pixel in those two layers, as it is combined, for digital image is requiring a process of some computational operators to give a fine double exposure image result. To analyze this process, the author applies Boolean algebra operations with a limitation to digital processing on binary images.**

***Keywords*—Binary image, Boolean algebra, digital image processing, double exposure.**

I. Introduction

Double exposure is a photography technique in which two photos are combined into one single image. In film photography, a double exposure image can be produced by exposing one film on top of other whereas in digital photography, it can be simply done by using Photoshop. Similarly to the practice of film photography, editing two digital image into a digital double exposure image is done by blending one layer after another, simply, and furthermore is by adjusting image hue and saturations for the background tones with the subject's colors [3].

Blending two digital images may cause confusion in which image should be put forward or else, backward. Observing a particular pixel on a digital image as result of two layers of digital images, the first layer has an image pixel representation so does the second layer, resulting as either a new pixel code or one of the two layers' code. This observation can be analyzed for its digital image processing.

This paper will use Boolean algebra to analyze the digital double exposure image processing, with limitation to binary image so that the pixel values only consisting of 1, representing white/light pixel and 0 as representation of black/dark pixel image color. The method used is by exposing layer over layers and observe the image produced with various Boolean algebra operations. Among those, one particular operations will give a fine double exposure image result.

II. Boolean Algebra Definition

Boolean algebra is defined in several abstract ways for that a particular structure that once was shown as a Boolean algebra, all properties of Boolean algebra in general apply to this particular structure. One of the most common way to define Boolean algebra is to specify structure constructing it and properties or axioms which that operations must satisfy [1]–[2], as is shown in Definition 1.

Definition 1. For a set B that is defined for two binary operators, $+$ and $\cdot$, and an unary operator, $'$, with 0 and 1 are elements of B, such that a tuple

$$< B,\ +,\ \cdot,\ ',0,1 >$$

is a Boolean algebra if these properties hold for all $a, b, c \in B$ (Huntington's Postulates with some modification):

1. Identity
   $a + 0 = a$
   $a \cdot 1 = a$
2. Commutative
   $a + b = b + a$
   $a \cdot b = b \cdot a$
3. Distributive
   $a \cdot (b + c) = a \cdot b + a \cdot c$
   $a + (b \cdot c) = (a + b) \cdot (a + c)$
4. Associative
   $(a + b) + c = a + (b + c)$
   $(a \cdot b) \cdot b = a \cdot (b \cdot c)$
5. Complement
   $a + a' = 1$
   $a \cdot a' = 0$

Note that 0 and 1 are two different elements in B. 0 is referred as the least element whereas 1 is the greatest element. These unique elements may differ for each Boolean algebra expressions, as in set equations are $\emptyset$ and U, also are $\boldsymbol{F}$ and $\boldsymbol{T}$ in propositions. Operator $+$ or OR correspond to $\vee$ and $\cup$, is denoting Boolean sums, as for $\cdot$ or AND corresponds to $\wedge$ and $\cap$, denoting Boolean products, and unary operator $'$ or NOT corresponds to $\backsim$ and $\overline{\ }$.

Similar to algebraic real number products, the symbol $\cdot$ can be deleted as long as it does not cause confusion leads to misinterpretation. Another similarity is shown on the rules of precedence for Boolean algebra where operations in parentheses are computed first followed by order of all complements, all Boolean products, and then all Boolean sums. However, there are still differences between Boolean algebra and algebra for

real number arithmetic those are stated below.

1. The first distributive law from Huntington's Prostulate, $a \cdot (b + c) = a \cdot b + a \cdot c$ has been known and can be applied on arithmetic algebra operations whereas the second distributve law, $a + (b \cdot c) = (a + b) \cdot (a + c)$, only can be applied on Boolean algebra and gives a wrong result when it is applied on arithmetic algebra.
2. There is no such thing as inverse for both multiplicative or sums, so that there is not division and subtraction in Boolean algebra which require inverse to compute.
3. Complements on the fifth axiom only does exist in Boolean algebra, also knows as negation.
4. Boolean algebra in general has undefined elements in the set B, as for two-valued Boolean algebra, B definition is limited to exactly only two values, 0 and 1.

In the representation of elements in Boolean algebra set B, variables such as a, b, or c are defined as symbols representating elements. With undefined values of elements in set B, to have a Boolean algebra it is needed to show

1. elements in B,
2. binary and unary operators, and
3. B as a set with the operators satisfy all five Boolean algebra properties.

III. Two-Valued Boolean Algebra

B is not specified to any number of elements, thus it is practically exists an infinite number of Boolean algebra. However, a Boolean algebra has at least two elements, refers to its definition of having two different elements.

Boolean algebra that is composed of elements with exactly two values, as it is, is two-valued Boolean algebra. It has various applications, typically in digital computation. Two-valued Boolean algebra is defined as a set of two elements, $B = \{0,1\}$ (its elements 0 and 1 is named *bit* – binary digit), two binary operators $+$ and $\cdot$, and an unary operator, $'$. The rules applied on its operators are as shown on these tables below.

Table I

a	b	$a + b$
0	0	0
0	1	1
1	0	1
1	1	1

Table II

a	b	$a \cdot b$
0	0	0
0	1	0
1	0	0
1	1	1

Table III

a	a'
0	1
1	0

With the rules stated above, the axioms in Definition 1 is satisfied as below.

1. Identity
   (i) $1 + 0 = 1$ and $0 + 0 = 0$
   (ii) $1 \cdot 1 = 1$ and $0 \cdot 1 = 0$
2. Commutative: as shown on the symmetry of binary operators Table I and Table II
3. Distributive: to show whether the rules satisfy this property, operations for at least three operands, $a, b, c \in B$ have to be presented on the truth table and see how distributive laws is satisfied
   (i) $a \cdot (b + c) = a \cdot b + a \cdot c$ (Table IV)
   (ii) $a + (b \cdot c) = (a + b) \cdot (a + c)$ (Table V)

Table IV

a	b	c	$b + c$	$a \cdot (b + c)$	$a \cdot b$	$a \cdot c$	$a \cdot b + a \cdot c$
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

Table V

a	b	c	$b \cdot c$	$a + (b \cdot c)$	$a + b$	$a + c$	$(a + b) \cdot (a + c)$
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

4. Associative: using truth table as presented in Table IV and Table V it can be shown that
   (i) $(a + b) + c = a + (b + c)$, and
   (ii) $(a \cdot b) \cdot c = a \cdot (b \cdot c)$.
5. Complement: from Table I, Table II, and Table III it can be known that the two-valued Boolean algebra satisfy
   (i) $a + a' = 1$, as for $a = 1$, $1 + 0 = 1$ and for $a = 0$, $0 + 1 = 1$, and
   (ii) $a \cdot a' = 0$, as for $a = 1$, $1 \cdot 0 = 0$ and for $a = 0$, $0 \cdot 1 = 0$.

It has been shown that the expression of $< B, +, \cdot, \ ' \ , 0,1 >$ with $B = \{0,1\}$ is a Boolean algebra, spesifically is a two-valued Boolean algebra. Furthermore, for every operations stated as Boolean algebra used in this paper will be specified as two-valued Boolean algebra without explicitly stating it.

Aside from the rules in Table I, Table II, Table III, and axioms in Definition 1, Boolean algebra has some identities for $a, b, c \in B$ as shown in Table VI below.

Table VI

Identity	Name
$(a')' = a$	Law of double complements
$a + a = a$ $a \cdot a = a$	Idempotent laws
$a + 0 = a$ $a \cdot 1 = a$	Identity laws
$a + 1 = 1$ $a \cdot 0 = 0$	Domination laws
$(a + b) + c = a + (b + c)$ $(a \cdot b) \cdot c = a \cdot (b \cdot c)$	Associative laws
$a \cdot (b + c) = a \cdot b + a \cdot c$ $a + (b \cdot c) = (a + b) \cdot (a + c)$	Distributive laws
$(a + b)' = a' \cdot b'$ $(a \cdot b)' = a' + b'$	De Morgan's laws
$a + a \cdot b = a$	Absorption laws

$a \cdot (a + b) = a$	
$a + a' = 1$	Unit property
$a \cdot a' = 0$	Zero property

IV. BOOLEAN FUNCTIONS

For there are values of Boolean function, there is Boolean expression that represents that Boolean function. As for that, any Boolean function can be represented by a Boolean sum of Boolean products of the variables and its complement, named Sum-of-Products (SOP) and Product-of-Sums (POS) respectively. It shows that for every Boolean function, it is possible to represent it using operators defined before, such as $+$, $\cdot$, and $'$. More advanced, a Boolean functions are not unique. It can be represented by another function equal to it with smaller set of operators.

A Boolean function can be denoted as following

$$f: B^n \rightarrow B$$

A. Minterm and Maxterm

As stated before, Boolean expression in which specifies a Boolean function can be represented into two forms, those are Sum-of-Products and Product-of-Sums. Expressions

$$f(x, y, z) = x'y'z + xy'z' + xyz$$

and

$$g(x, y, z) = (x + y + z)(x + y' + z)(x + y' + z')(x' + y' + z)$$

are equal, and its equality can be shown in a truth table. Function f is an expression represented in SOP form whereas function g is representing Boolean function in POS form. Each term in each function consists of a complete literal by variables $x, y,$ and z, be it in its complement or not. Terms are divided into two kinds of term, minterm for the result of products and maxterm for the result of sums.

Generally, elements in Boolean function with n variables $x_1, x_2, x_3, \ldots, x_n$ is a minterm if it is represented as

$$\widetilde{x_1} \cdot \widetilde{x_2} \cdot \widetilde{x_3} \cdot \ldots \cdot \widetilde{x_n}$$

and is a maxterm if it is represented as

$$\widetilde{x_1} + \widetilde{x_2} + \widetilde{x_3} + \cdots + \widetilde{x_n}$$

with $\widetilde{\ }$ is a notation for that each x_i can be in its complement or not.

B. Boolean Functions Simplification

Boolean expressions of a Boolean function are not unique, therefore there must be any way to simplify a Boolean expression which is known as function minimization. In example, a function $f(x, y) = x'y + xy' + y'$ can be simplified into a function $f(x, y) = x' + y'$.

There are various ways in order to get a simpler function from a presented Boolean function, such that stated below.

1. Boolean function implification by algebra.
2. Karnaugh's map Method.
3. Quine-McCluskey Method.

C. Karnaugh's Map Method

Karnaugh's map is a map which is formed of squares, adjacently, for each square is representing a minterm. For squares are side by side if and only if the minterms represented in it is differ by 1.

1. Two Variables

   Let two variables in a Booloan function be x and y, so that will be formed a 2×2 Karnaugh's map with the the first row identified by 0 (representing x') whereas the second row identified by 1 (representing x), and the first column identified by 0 as y' representation where the second row is identified by 1 (representing y). Each square representing a minterm of the intersection. Commonly, Karnaugh's map is formed as below.

		y	
		0	1
x	0	$x'y'$	$x'y$
	1	xy'	xy

2. Three Variables

   Let three variables in a Boolean function be x, y, adn z, so that will be formed a 2×4 Karnaugh's map with the first and second row the same as the one in two variables, and as for column it is by order identified by 00, 01, 11, 10, with each 0 represents the complement and 1 represents the non-complement of y followed by z. Each square representing a minterm of the intersection so the Karnaugh's map is formed as below.

		yz			
		00	01	11	10
x	0	$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$
	1	$xy'z'$	xy'z	xyz	xyz'

   In this three variables Karnaugh's map it is shown how each adjacent square is only differ by 1 that is why the column is in 11 and then 10 order eventhough if read as binary number it is normally be 10 followed by 11. Becase there is only one different bit between 01 and 11 whereas 01 and 10 have two differebt bits.

3. Four Variables

   Let four variables in a Boolean function be w, x, y, and z, so that will be formed a 4×4 Karnaugh's map. The principle is similar to the columns in the Karnaugh's map with three variables by having wx for the rows and yz for the columns. Each square representing a minterm of the intersection.

4. Function Minimization Technique with Karnaugh's Map

   The usage of Karnaugh's map in simplifying a Boolean function is done by grouping every 1-valued square that has 1-valued square by its side. The group of 1-valued square can be either dual (two), quad (four), or octet (eight), or the combination of it. The explaination for this technique is better by examples as presented below.

   (i) Dual

   A Karnaugh's map for Boolean function

$$f(w,x,y,z) = wxyz + wxyz'$$

is shown below.

		yz			
		00	01	11	10
wx	00	0	0	0	0
	01	0	0	0	0
	11	0	0	1	1
	10	0	0	0	0

It can be seen on the Karnaugh's map presented that this Boolean function has the same literal for w, x, and y identified by 1, therefore the new simplified function equal to this Boolean function is

$$f(w,x,y,z) = wxy$$

(ii) Quad

The same method will be applied for a Boolean function containing quad 1-valued square as shown in the Karnaugh's map below.

		yz			
		00	01	11	10
wx	00	0	0	0	0
	01	0	0	0	0
	11	1	1	1	1
	10	0	0	0	0

This Karnaugh's map is representing Boolean function

$$f(w,x,y,z) = wxy'z' + wxy'z + wxyz + wxyz'$$

is simplified into

$$f(w,x,y,z) = wx$$

Observed as duals, the grouping can be done as this Karnaugh's map shown below.

		yz			
		00	01	11	10
wx	00	0	0	0	0
	01	0	0	0	0
	11	1	1	1	1
	10	0	0	0	0

$$\begin{aligned} f(w,x,y,z) &= wxy' + wxy \\ &= wx(y' + y) \\ &= wx(1) \\ &= wx \end{aligned}$$

Another form of quad is shown as below.

		yz			
		00	01	11	10
wx	00	0	0	0	0
	01	0	0	1	1
	11	0	0	1	1
	10	0	0	0	0

A Boolean function

$$g(w,x,y,z) = w'xyz + w'xyz' + wxyz + wxyz'$$

is simplified into

$$g(w,x,y,z) = xy$$

(iii) Octet

For a Boolean function with Karnaugh's map as presented below.

		yz			
		00	01	11	10
wx	00	0	0	0	0
	01	1	1	1	1
	11	1	1	1	1
	10	0	0	0	0

$$\begin{aligned} f(w,x,y,z) = {} & w'xy'z' + w'xy'z + w'xyz + w'xyz' \\ & + wxy'z' + wxy'z + wxyz + wxyz' \end{aligned}$$

is simplified into

$$f(w,x,y,z) = x$$

5. Exclusive OR (XOR)

A Boolean function of three variables

$$f(x,y,z) = x \cdot y' + x' \cdot y$$

is shown in the Karnaugh's map representation as below.

		yz			
		00	01	11	10
x	0	0	1	1	0
	1	1	0	0	1

The equal function f after simplification is

$$f(x,y,z) = x'z + xz'$$

Furthermore, this function is known as exclusive OR with operator symbol $\oplus$.

IV. Digital Image Processing

A. Image Processing Operations

There are several steps to process a digital image. It begins with capturing the image itself which related to optical system. This step requires technical and scientific application to get a 2-D and 3-D formation. At this point, image has been sensed and is needed to be brought into a form which digital computers can treat, or called digitization. Digitization is done by image analysis. It is a process of obtaining numerical data from images. Furthermore, as the image is now in digital form, the next steps ahead are advanced steps such as segmentations to distinguish the object of interest from the background, texture analysis by reading pattern, and even motion [4]–[5].

B. Binary Images

Binary images are basically black-and-white images. It is important to keep in mind that black-and-white images and gray scale images are two different form of images. Gray scale images are composed of pixels with unsaturated color whereas black-and-white images are practically images composed of only black or white pixels.

To convert a gray scale or color original image to binary images can be done by thresholding. It is to distinguish the object of interest from its background by some kind of measurement procedure. In addition, two or more binary images can be combined by operating each pixel following the rules as presented in Boolean algebra.

V. ANALYSIS

Having two binary images, Fig. 1 and Fig. 2 to produce a double exposure image.

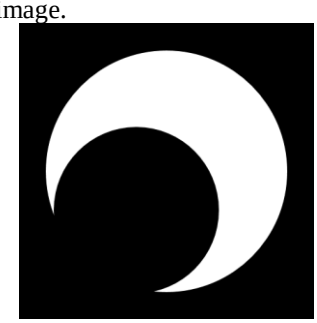


Fig. 1 A binary image A

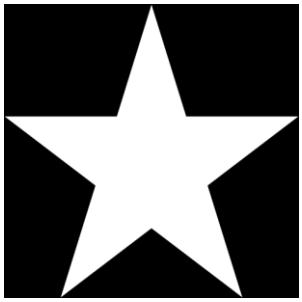
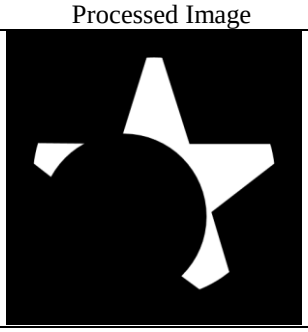
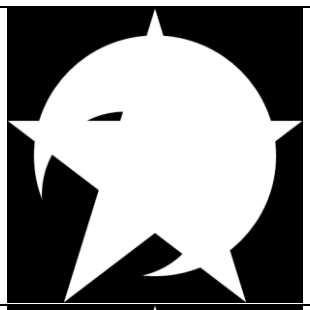
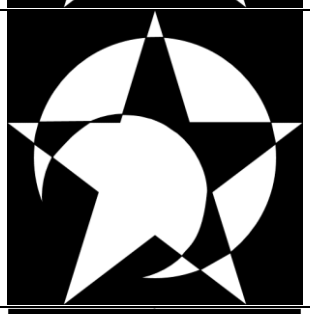
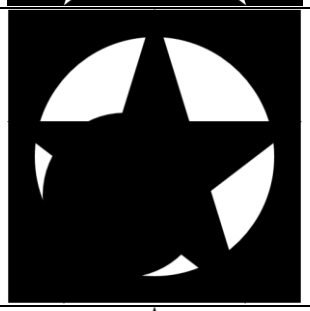
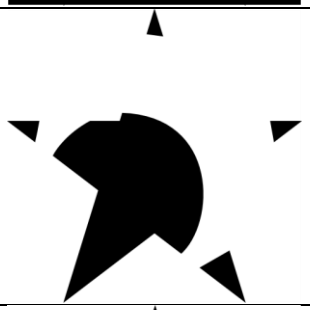
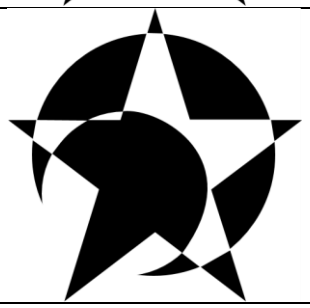


Fig 2. A binary image B

By defining all pixels in white color as 1, as it is the feature of each binary image, and the rest pixels in black color as 0, meant to be the background. Using the rules presented on Table I, Table II, and Table III each binary combination gives different binary image result as presented in the table below. White denoted as w and black denoted as b.

Table VII

Boolean Combination	Processed Image
$A\ AND\ B$ $1 \cdot 1 = 1 \sim w$ over $w = w$ $1 \cdot 0 = 0 \sim b$ over $w = b$ $0 \cdot 1 = 0 \sim w$ over $b = b$ $0 \cdot 0 = 0 \sim b$ over $b = b$	
$A\ OR\ B$ $1 + 1 = 1 \sim w$ over $w = w$ $1 + 0 = 1 \sim b$ over $w = w$ $0 + 1 = 1 \sim w$ over $b = w$ $0 + 0 = 0 \sim b$ over $b = b$	
$A\ Ex-OR\ B$ $1 \oplus 1 = 0 \sim w$ over $w = b$ $1 \oplus 0 = 1 \sim b$ over $w = w$ $0 \oplus 1 = 1 \sim w$ over $b = w$ $0 \oplus 0 = 0 \sim b$ over $b = b$	
$A\ AND\ NOT-B$ $1 \cdot 1 = 1 \sim b$ over $w = w$ $1 \cdot 0 = 0 \sim w$ over $w = b$ $0 \cdot 1 = 0 \sim b$ over $b = b$ $0 \cdot 0 = 0 \sim w$ over $b = b$	
$A\ OR\ NOT-B$ $1 + 1 = 1 \sim b$ over $w = w$ $1 + 0 = 1 \sim w$ over $w = w$ $0 + 1 = 1 \sim b$ over $b = w$ $0 + 0 = 0 \sim w$ over $b = b$	
$A\ Ex-OR\ NOT-B$ $1 \oplus 1 = 0 \sim b$ over $w = b$ $1 \oplus 0 = 1 \sim w$ over $w = w$ $0 \oplus 1 = 1 \sim b$ over $b = w$ $0 \oplus 0 = 0 \sim w$ over $b = b$	

Comparing this process with the process of creating double exposure image in film photography, there are various ways to get there. Two of them are by planning multiple exposure and by *sandwiched* negatives.

In planning multiple exposures, the order is the most important as the successive images will *mark* the film. Exposing film much more affected by light than dark. Succesive layer exposed on top of the previous layer. Light re-exposed over light will be resulting of light, but note that it is not the light replacing light on negative, instead, it gives very little or no effect to the negative. As for light over dark will also produce light on

negative with the light on the second exposure will affect the negative normally as it is. As the negative's sensitivity is still in partially used condition, the successive exposures can build up light levels up until all the sensitivity is used. Simplifying the exposures to only dip light and dip dark, this process can be modelled as Boolean operations below with light denoted as 1 and dark as 0.

$$1\ (opr) 1 = 1$$
$$1\ (opr)\ 0 = 1$$
$$0\ (opr)\ 1 = 1$$
$$0\ (opr)\ 0 = 0$$

It shows that Boolean algebra operation appropriate for replacing (opr) in the equations above is $+$ or denoted as OR.

Another method to produce double exposure is by sandwiching the negatives as it is proceed for printing. The advantage of this method is on how photographer can decide how the two negatives will be positioned for blending.

VI. Conclusion

Double exposure as one of the photography technique gives a magical image processed from two images with different exposures. Producing a double exposure image on digital process can be done by operating each pixel that overlays. This operation, then is analyzed using Boolean algebra by having a limitation to image processing on binary image. With Boolean combinations, two binary images processed into various type of new blended images. Comparing the digital process to film photography process in producing double exposure images, it has come to conclusion that the Boolean combination which give a fine double exposure image result is to combine using $+$ or OR operator.

VII. Acknowledgment

The author is highly grateful to Allah ﷻ, for all the blessings in which only with that author can finish this paper. Also to Discrete Mathematic lecturer at Informatics Engineering program class K-01, Dr. Ir. Rinaldi Munir, M. T. for all the knowledges he has given to his students with dedication, that gave the author positive experience both related to the lecture and the insight about everything. Author's gratitude is for her family, friends, classmates, and other parties, for their support be it direct or indirectly to the fine process of this paper work.

References

[1] R. Munir, Matematika Diskrit, Bandung: Penerbit Informatika, ed. 4, 2010, ch. 7.
[2] K. H. Rosen, Discrete Mathematics and Its Applications, 7th Edition, New York: The McGraw-Hill, 2012, ch. 12.
[3] S. Covey, Photoshop for Artists: A Complete Guide for Fine Artists, Photographers, and Printmakers, First Edition, New York: Watson-Guptill, 2012.
[4] https://micro.magnet.fsu.edu/primer/digitalimaging/russ/index.html accessed on December 9th 2018, 18.48 (GMT+7).
[5] B. Jähne, Digital Image Processing, 6th revised and extended edition, Berlin: Springer-Verlag, 2005.
[6] G. Rand and D. Litschel, Black & White Photography, Second Edition, New York: Thomson Learning, 2002, ch. 11.

Pernyataan

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 9 Desember 2018

Fitria Budi Ananda
13517133