The Usage of Decision Tree in Medical Field

Dicky Adrian - 13516050[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13516050@std.stei.itb.ac.id

Abstract*— Many instances in the world now have some kind of big data. With big data, we also need to know how to process them. One of the way to process big data is constructing a tree. Some instances with a big data is hospital, or to be more generalized medical instances. In medical instances there will be time when we need to make a quick decision. A decision tree can make a quick decision using some algorithm and with decision trees, risk can be reduced. One example is the caesarian procedure for pregnant women, although in these years the caesarian procedure has become more safe, but it is still a procedure that have some risk. With the database and the decision tree, we can generalize what is the best course of action depending on the situation, so the risk will be reduced. And it's proven with the success rate over 80%.

***Keywords*— Decision Tree, Medical, Tree**

I. INTRODUCTION

In the recent years, databases have become more and more advanced than ten years ago. That makes some trouble in handling cases with big database. With that much knowledge, we need to know how to determine a decision that is optimal for every case that may occur. Decision tree is one way to decide what is the best possible decision to make in each case.

Decision Tree can be used in many ways, from the simplest case to advanced cases. For example, decision tree can be used to determine whether or not to go outside based the humidity, raining, and wind parameters, or where is the best place to go in this time of the year for a holiday.

Other field which benefit from decision trees are the one with big database. For example, medicine, doctors need to know fast how to handle patients with some symptoms, and sometimes, there are not enough doctors in the hospital. That's when a machine is needed to take the job. Other than that, a decision tree can also determine what drugs must be taken based on the symptoms.

One of the procedure that needs attention is caesarian section for pregnant women. The rate of mortality of pregnant women is not stable, it's always fluctuating. One way to stabilize the mortality rate is to have a system that can always determine the best decision for each patient. But, of course the procedure must always be accompanied with a certified doctor, since we can't depend fully on the machine itself.

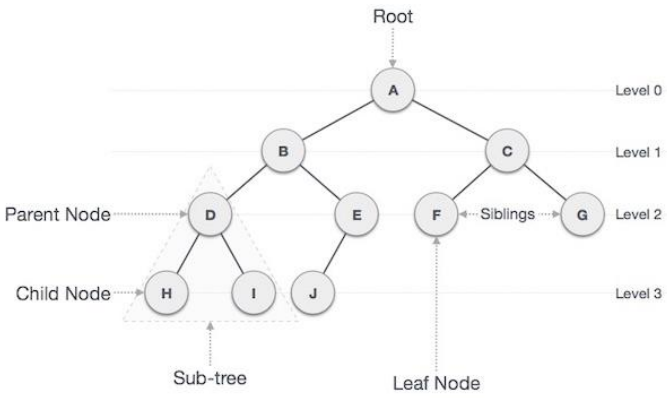
To make a decision tree, there are some algorithms around the world. The easy one is effective enough, but can't handle some types of data. But, there are also other algorithms which can handle most data types but comes with a higher "price". So, as engineers it's really important to know how to implement trees in the most effective way and the most efficient way.

II. TREE

A. Definition

Tree in data structure means an undirected graph that contains no circuit. Tree is a collection of data (Node) which is organized in hierarchical structure and this is a recursive definition [7]. There are two types of trees, first is free tree. Free tree is a tree with no terminologies such as root, leaf, etc. There is also a rooted tree, which contain root, child, parent, etc. All of these terminologies will be explained later. When there are two or more trees in one case, it can be called a "forest".

B. Terminology



Source: https://www.tutorialspoint.com/data_structures_algorithms/tree_data_structure.htm

There are a lot of terminology in tree data structure. The terminologies are:

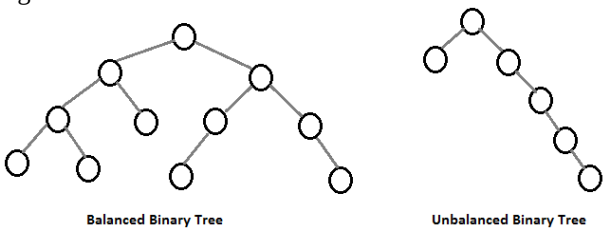
- Root
  The node at the top of the tree, in the image the root of the tree is 'A'.
- Child and Parent
  A child is a node which pointed by the parent node.
  For example, 'D' is the child of 'B' and 'B' is the parent of 'D'
- Path
  A path is how one node can go to another node with the help of the edges. For example, the path from 'A' to 'H' is A, B, D, H, and the length of the path is 3.
- Descendant and Ancestor
  If there is a path between 'x' and 'y' nodes, then 'x' is the ancestor of 'y' and 'y' is the descendant of 'x'. For

example, in the image, ‘H’ is the descendant of ‘B’ and ‘B’ is the ancestor of ‘H’.

- Sibling
  If one parent has 2 or more nodes, then if ‘X’ and ‘Y’ are the children from ‘Z’, then ‘X’ and ‘Y’ are siblings.
- Level
  Level represents the generation of the nodes, from the image it’s clear that the node ‘B’ is level 1, and so on.
- Height
  Height represents the max level of the tree,
- Leaf
  A leaf in tree data structure is a node that doesn’t have any child for example the node ‘H’, ‘I’, ‘J’, ‘F’ and ‘G’.
- Internal Nodes
  Internal nodes unlike leaf, is the node in the tree that at least have one child. For example, ‘A’, ‘B’, ‘C’, ‘D’ and ‘E’.
- Sub tree
  A sub tree represents the descendants of a node.

C. Types of Trees

- Balanced Tree
  Balanced tree is a tree where there are no leaf on the right side which is farther than the leaf on the left side.

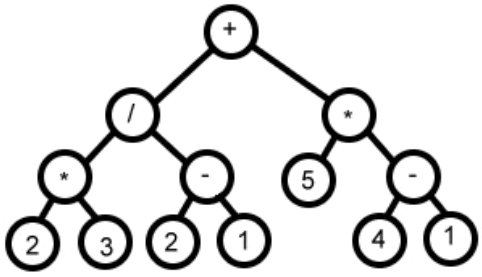


  Source: https://www.ocf.berkeley.edu/~shidi/cs61a/w/images/8/88/Balanced_vs_unbalanced_BST.png
- N-Ary Tree
  N-Ary tree is a rooted tree and every node on that tree can’t have more than *N* children. If every node except the leaf on the tree have approximately *N* children, then the tree is called full N-ary tree.
- Binary Tree
  A binary tree is a tree and every node on that tree can’t have more than 2 children, in other words, a binary tree is a 2-ary tree.
- Ordered Tree
  Ordered tree is a tree where the children of every parent is sorted in some way.

D. Application of Binary Tree

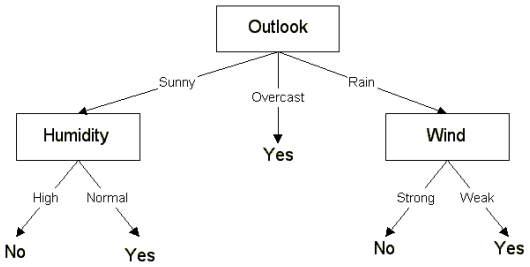
Binary tree is a very important data structure in computer science. It’s a powerful data structure that can do many things.

- Expression Tree
  An expression tree is a tree that contains a mathematical expression such as subtraction, addition, multiplication, etc. Nodes can contain both operand and operators. Nodes with operators will operate the children of that node. For example, given a tree below



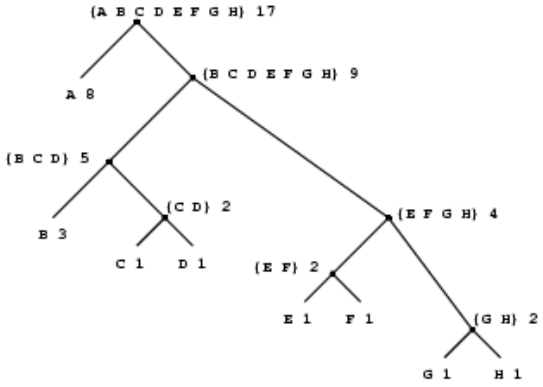
Source: https://i.stack.imgur.com/8QFVk.gif
The evaluation of the tree above is ((2*3)/(2-1))+(5*(4-1)).

- Decision Tree
  A decision tree is a tree that contains a solution on each leaf. The root and each internal node are labeled with question. A decision tree doesn’t always come in a binary tree form, it can also be N-ary tree with *N* greater than two.



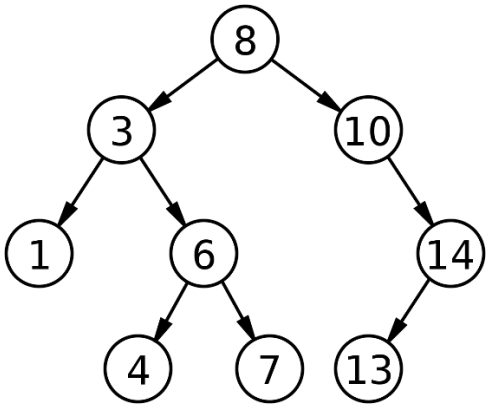
Source: https://www.cise.ufl.edu/~ddd/cap6635/Fall-97/Short-papers/Image3.gif

- Prefix code
  Prefix code is a set of codes, which every member of the set is not a prefix of other member. For example:
  - {000, 001, 01, 10, 11} is a prefix code
  - {1, 00, 01, 000, 0001} is not a prefix code because ’00’ is the prefix fore ‘000’ and ‘0001’
- Huffman Code
  Huffman code is one algorithm to shorten the length of bitwise string based on how frequent the letter shows up. More often the letter shows up, the algorithm will make the bitwise size smaller. This kind of algorithm is called compression algorithm. The steps to get Huffman code are:
  1. Pick two symbols with the least probability, then make the two of them into two nodes for one parent which is the concatenation of the two.
  2. The concatenation of the two previous symbol is a new symbol and the two symbol picked before is now “erased”.
  3. Repeat step 1 until all the symbol “erased”.



Source: https://mitpress.mit.edu/sicp/full-text/sicp/book/chapter-2/figs/huffman.gif

- Binary Search Tree
  A binary search tree is one of the most important tree in data structure. It can handle the fundamentals such as searching, inserting elements, deleting elements. The work of binary search tree is determined by the key of every node. Left subtree's key is less than the root's key and the right subtree's key is greater than the root's key. This occur on every node.



Source: https://upload.wikimedia.org/wikipedia/commons/thumb/d/da/Binary_search_tree.svg/1200px-Binary_search_tree.svg.png

III. CONSTRUCTING DECISION TREE

A. Stages

There are three main phase to construct a decision tree:

1. Construction Phase
   In this phase, the tree is constructed with the entire database. It is constructed by doing recursion technique until a stopping criteria is met.
2. Pruning Phase
   After constructing the tree, there's a chance that the tree is not in the best version due to over-fitting. In this phase, we make the previously built tree more effective by removing some of the lower branches and nodes to improve the performance of the tree.
3. Processing the pruned tree
   This stage is an optional stage where we improve the tree to make the tree more understandable.

B. Algorithm

Some of the algorithm are to construct a decision tree are:

1. ID3 Algorithm
   This Algorithm first introduced by J.R. Quinlan and it is one of the simplest algorithm to construct a decision tree. This algorithm process the database top-down until it create a tree and no backtracking. ID3 Algorithm uses *entropy* and *information gain* to construct a decision tree. But, ID3 Algorithm doesn't do any pruning, which result in not very effective decision tree.
   As stated above, we need to know the entropy of the database we have. Entropy calculates the homogeneity of a sample. If the sample is homogenous then the entropy will be 0, and that sample is a leaf node. The steps are:
   - Calculating entropy
     Construct a frequency table of one attribute, for example 'play tennis' is

Play Tennis	
Yes	No
9	5

The entropy is calculated by the equation:

$$E(S) = \sum_{i=1}^{c} -p_i log_2 p_i$$

On the example above we will get:

$$E(play\_tennis) = E(5,9)$$
$$E(play\_tennis) = E(0.36, 0.64)$$
$$E(play_{tennis}) = -(0.36\ log_2 0.36) - (0.64\ log_2 0.64)$$
$$= 0.94$$

If there are more than one attribute then we calculate it as

$$E(T,X) = \sum_{c \in X} P(c)E(c)$$

For example,

		Play Tennis		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14

$$E(PlayTennis, Outlook) = P(Sunny)\ E(3,2) + P(Overcast)\ E(4,0) + P(Rainy)\ E(2,3)$$

   - Calculating Gain
     Gain is obtained by calculating the deviation on each predictor's entropy and other predictor's entropy.

After that, first, we decide which node is the target of the decision tree, then we calculate the gain on each

predictor. Choose the one with the most gain as a decision node, then divide the dataset by its branches, repeat the same process on every branch.

However, as stated above, this algorithm has some disadvantages, such as:

- ID3 doesn't produce the most optimal solution because the algorithm doesn't do any backtracking on each branch.
- ID3 can overfit to the training data.
- ID3 can only make trees from nominal data, not continuous data.

2. C4.5 Algorithm

   C4.5 algorithm is also introduced by J.R. Quinlan as the evolution of ID3 Algorithm. As we know before, the ID3 algorithm has some disadvantages, in the C4.5 algorithm, they are fixed. The C4.5 algorithm uses gain ratio as splitting criteria and uses error-based pruning after the growing phase.

   The most important updates from ID3 Algorithm to C4.5 algorithm are:

   - C4.5 algorithm uses pruning procedure, so some branches that don't contribute to accuracy are replaced by leaf nodes.
   - C4.5 allow some values to be missing
   - C4.5 can handle not only nominal data, but also continuous data.

   There is another update to the C4.5 algorithm which is the C5.0. C5.0 algorithm claims to have more efficiency both in memory and computation time.

3. CART Algorithm

   CART or Classification and Regression Trees was created by Breiman *et al.*(1984). In this algorithm, the tree is a binary tree which has exactly two outgoing edges. The splits are selected by Twoing Criteria and it also uses pruning, with Cost-Complexity Pruning. As the name suggest, CART Algorithm can also make regression trees. In the regression trees, the leaves predict a real number and not a class.

4. CHAID

   CHAID which stands for Chi-squared-Automatic-Interactive-Detection is another algorithm to construct a decision tree. This algorithm will find the least significantly difference attribute from the target attribute. The attributes are measured by some tests, an F test is used is the target attribute is continuous, a Pearson chi-squared test if it is nominal and a likelihood ration test if it is ordinal.

   For each input attribute, the algorithm will check if the value is greater than a certain merge threshold. If it is higher, then the attributes are merged. The method of splitting is by making a group of homogenous values the children of a node. This procedure stops if a maximum tree depth is reached, maximum number of cases in a node for being a parent is reached and a minimum number of cases in a node for being a child node is reached.

5. QUEST

   QUEST which stands for The Quick, Unbiased, Efficient Statistical Tree algorithm is an algorithm that can handle univariate and linear combination splits. This algorithm uses several test, such as ANOVA F-test or Levene's test or Pearson's chi-square test. First the algorithm will calculate ANOVA F-statistic for each attribute, and the one with the largest value will be selected as a splitting node. Otherwise, Levene's test will be performed and if the largest value of the test is greater than a certain threshold, the node is selected as a splitting node.

   This algorithm also uses Quadratic Discriminant Analysis to find the optimal splitting point for each input attribute, and uses ten-fold cross-validation to prune the tree.

6. Other Algorithms

   There are a lot of other algorithms to create Decision trees, some are CAL5, FACT, LMDT, T1, PUBLIC, and MARS. Each of them has their own uses, depending on the type of data. They also have their own weaknesses and strength. The CAL5 is designed specifically for numerical-valued attributes, the FACT algorithm is an earlier version of QUEST, the LMDT algorithm construct a decision tree by using multivariate tests, the T1 algorithm a one level decision tree that classifies instances using one attribute, the PUBLIC uses MDL cost to prune the decision tree in order to reduce computational complexity, the MARS uses multiple regression to create the decision tree.

D. Tree Representation in Data Structure

Tree can be represented in many ways, two of them are list representation and Left-child Right-sibling representation. In list representation, it uses two types of nodes, one list contain the data of the node, and one contain only references. Each node is connected to each other by looking at the pointer in the reference nodes.

The second one is more understandable, every element in the list has two pointers, one to the left child, and the other to the right sibling. If there is no pointer to the left child or the right sibling, the pointer will point to NULL. This occur on each node on the tree.

E. Advantages and Disadvantages of Decision trees

As a data structure, trees also have their advantages and disadvantages. Some of the advantages of decision trees are:

1. Decision trees are easy to understand, because they are self-explanatory.
2. Decision tree can handle nominal and numeric input attributes
3. Decision tree representation is enough to represent discrete value
4. Decision trees can handle missing cases
5. Decision trees can handle error cases
6. Decision trees are considered to be a non-parametric method because they do not include any assumptions about the space distribution and on the classifier structure.

7. If the classification cost is high, trees can be used because they ask only for the values of the features along a single path.

Some disadvantages of decision trees are:

1. With some algorithm, the result will always be discrete values
2. They tend to only perform well if a few highly relevant attributes exist.
3. Some fragmentation problem causes partitioning data into smaller fragments
4. To handle missing cases, the algorithm must employ special mechanism to process the missing value
5. Decision trees are very unstable, a minor change will change everything from the root to the leaves

IV. DATA AND CALCULATION

As stated in chapter I, there are usage of decision trees in medical field such as determining whether a pregnant women need a caesarian procedure or not. Caesarian procedure itself has some risk, such as, the baby's immune system is not as strong at first so the death rate of the baby is higher than normal procedure. But in some case, a pregnant women need caesarian procedure to save both the mother and the baby. In this data, from a paper written by Farhad S. G., Peyman Mohammadi, and Parvin Hakimi, they choose five attributes to construct the decision tree. They are age, number of pregnancy, delivery time, blood pressure and heart status, and to construct the tree they use the C4.5 algorithm so it's more efficient.

No	Age	Pregnancy NO	Delivery Time	Blood Presure	Heart Problem	Caesarian
1	22	1	Timely	High	apt	No
2	26	2	Timely	Normal	apt	Yes
3	26	2	Premature	Normal	apt	No
4	28	1	Timely	High	apt	No
5	22	2	Timely	Normal	apt	Yes
6	26	1	Premature	Low	apt	No
7	27	2	Timely	Normal	apt	No
8	32	3	Timely	Normal	apt	Yes
9	28	2	Timely	Normal	apt	No
10	27	1	Premature	Normal	apt	Yes
11	36	1	Timely	Normal	apt	No
12	33	1	Premature	Low	apt	Yes
13	23	1	Premature	Normal	apt	No
14	20	1	Timely	Normal	inept	No
15	29	1	Latecomer	Low	inept	Yes
16	25	1	Latecomer	Low	apt	No
17	25	1	Timely	Normal	apt	No
18	20	1	Latecomer	High	apt	Yes
19	37	3	Timely	Normal	inept	Yes
20	24	1	Latecomer	Low	inept	Yes
21	26	1	Premature	Normal	apt	No
22	33	2	Timely	Low	inept	Yes
23	25	1	Premature	High	apt	No
24	27	1	Timely	Low	inept	Yes
25	20	1	Timely	High	inept	Yes
26	18	1	Timely	Normal	apt	No
27	18	1	Premature	High	inept	Yes
28	30	1	Timely	Normal	apt	No
29	32	1	Timely	High	inept	Yes
30	26	2	Premature	Normal	inept	No
31	25	1	Timely	Low	apt	No
32	40	1	Timely	Normal	inept	Yes
33	32	2	Timely	High	inept	Yes
34	27	2	Timely	Normal	inept	Yes
35	26	2	Latecomer	Normal	apt	Yes
36	28	3	Timely	High	apt	Yes
37	33	1	Premature	Normal	apt	No
38	31	2	Latecomer	Normal	apt	No
39	31	1	Timely	Normal	apt	No
40	26	1	Latecomer	Low	inept	Yes
41	27	1	Timely	High	inept	Yes
42	19	1	Timely	Normal	apt	Yes
43	36	1	Premature	High	apt	Yes
44	22	1	Timely	Normal	apt	Yes
45	36	4	Timely	High	inept	Yes
46	28	3	Timely	Normal	inept	Yes
47	26	1	Timely	Normal	apt	No
48	32	2	Timely	High	inept	Yes
49	26	2	Latecomer	Normal	apt	No
50	29	2	Timely	Low	inept	Yes
51	33	3	Latecomer	Normal	inept	No
52	21	2	Premature	Low	inept	Yes
53	30	3	Latecomer	High	apt	No
54	35	1	Premature	Low	apt	No
55	29	2	Timely	Normal	inept	Yes
56	25	2	Timely	Normal	apt	No
57	32	3	Premature	Low	inept	Yes
58	21	1	Timely	Low	apt	Yes
59	26	1	Timely	High	apt	Yes
60	30	2	Premature	High	inept	Yes
61	22	1	Latecomer	High	apt	No
62	19	1	Timely	Normal	apt	Yes
63	32	2	Timely	Low	apt	Yes

64	32	2	Timely	Normal	inept	Yes
65	31	1	Latecomer	High	inept	No
66	35	2	Timely	Normal	apt	Yes
67	28	3	Timely	Normal	apt	Yes
68	29	2	Timely	Normal	inept	No
69	25	1	Timely	Low	apt	Yes
70	27	2	Latecomer	Low	apt	No
71	17	1	Timely	Low	apt	Yes
72	29	1	Latecomer	Low	inept	Yes
73	28	2	Timely	Normal	apt	No
74	32	3	Timely	Normal	inept	No
75	38	3	Latecomer	High	inept	Yes
76	27	2	Premature	Normal	apt	No
77	33	4	Timely	Normal	apt	Yes
78	29	2	Premature	High	apt	Yes
79	25	1	Latecomer	Low	apt	Yes
80	24	2	Latecomer	Normal	apt	No

Source: http://research.ijcaonline.org/volume52/number6/pxc3881613.pdf

From the data, they collected, as stated before, they uses the C4.5 algorithm to construct the decision tree due to its capabilities and the efficiency of the algorithm. In the algorithm they construct a tree with the size of 31 and it has 21 leaves node. The algorithm is a top-down, greedy search procedure. After constructing the tree it can be seen that most women with inept heart status didn't have a natural delivery and over 65% of those inept heart case savors an abnormal pressure of blood

After they construct the tree, they evaluate the data once again this time with the tree, and the result is pretty good, the tree can guess most cases correctly. Out of 80 cases, the tree guess 69 cases correctly, or approximately 86.25% of the time correctly. This means that with the decision tree, we can make medical guesses efficiently and correctly under normal circumstances. Even though, it's not 100% accurate, but there can be some improvements in the future that make the decision trees more accurate. Of course, if there are some abnormalities in the patient's body, it's best to ask a doctor rather than trusting a machine to decide a surgery, because until now, machine still can't handle the abnormalities in data.

V. CONCLUSION

Medical field has one of the biggest database in the world. How to treat each patient is determined with each patient's unique condition. In order to make things more efficient we can simplify the problem with a decision tree. With the tree, we can understand what is the most important thing in deciding the best course of action. For example, in caesarian procedure for delivering a baby, the mother's heart status is the most important status. With this information, we can make decision faster and accurately.

There are still other uses for decision tree in many field, and medicine is one of them. Other field that benefit from decision tree are business, economy, chemical (drug), and many more. Everything that needs classification can be done with a decision tree.

VII. ACKNOWLEDGMENT

This paper was supported by Institut Teknologi Bandung. I thank all of my references' writers who provided insight and expertise that greatly assisted the research. I also hope that I didn't step any boundaries while writing this paper.

I would also like to express my special thanks to all the lecturers that made this paper possible, Dra. Harlili, M.Sc. and also with the help of all the friends that helped me during the process of writing this paper. I would also like to express my thanks to my parents for the financial and moral support.

REFERENCES

[1] Dr. Rinaldi Munir, "Diktat Kuliah Matematika Diskrit" 4th ed
[2] https://xlinux.nist.gov/dads/HTML/tree.html accessed at 2nd December 2017, 22.02
[3] https://www.tutorialspoint.com/data_structures_algorithms/tree_data_structure.htm accessed at 2nd December 2017, 22.07
[4] https://www.ijltet.org/wp-content/uploads/2012/10/23.pdf accessed at 2nd December 2017, 23.30
[5] https://doc.lagout.org/Others/Data%20Mining/Data%20Mining%20with%20Decision%20Trees_%20Theory%20and%20Applications%20%282nd%20ed.%29%20%5BRokach%20%26%20Maimon%202014-10-23%5D.pdf accessed at 3rd December 2017, 00.17
[6] http://research.ijcaonline.org/volume52/number6/pxc3881613.pdf accessed at 3rd December 2017, 00.18
[7] http://btechsmartclass.com/DS/U3_T1.html accessed at 3rd December 2017, 09.42
[8] http://btechsmartclass.com/DS/U3_T2.html accessed at 3rd December 2017, 10.02
[9] http://www.saedsayad.com/decision_tree.htm accessed at 3rd December 2017, 00.42

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, 3 Desember 2017

Dicky Adrian – 13516050