Association Rule Analysis using CT-Pro and Hash-based Algorithm in Violence Case of Children

Amir Hamzah Siregar[1], Maya Silvi Lydia[1] and Sutarman Wage[2]
[1]*Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Medan, Indonesia*
[2]*Faculty of Mathematics and Natural Sciences, Universitas Sumatera Utara, Medan, Indonesia*

Keywords: Association Rule Mining, CT-Pro, Hash-Based, Frequent Itemset.

Abstract: The searching technique for frequent itemset patterns in finding support and confidence values with the a priori algorithm association rule method has a weakness in performance (because it has to read the database repeatedly in determining frequent itemset). This becomes a serious problem if the database is large, reading the database repeatedly results in very high processing times for a long time to generate support & confidence values. A special approach in analyzing association rules using CT-Pro and Hash-Based is needed. CT-Pro has a CFP-Tree data structure that allows a faster search for frequent itemset where the number of paths or trees that are built was compressed. Hash-based works with a hashing technique where the database was only read in the first iteration by entering the candidate itemset in the hash table. The test results were carried out with 3% support and 15% confidence, CT-pro formed 22 rules and an execution time of 0.25 seconds, while Hash-based formed 22 rules and an execution time of 0.75 seconds. A new pattern of crime that was found with the highest confidence and support was when an act of sexual harassment resulted in physical torture with a confidence of 59%, a support count of 34, and a lift ratio of 1.29.

1 INTRODUCTION

In the law number 23 of 2002 regulates the protection of children (someone under 18 years of age). Violence perpetrated against children is behavior that is abusive either by parents or adults. Based on data from the Office of Women's Empowerment and Child Protection of North Sumatra Province, the P2TP2A Unit (Integrated Service Center for the Empowerment of Women and Children) states that the total number of violence against children in 2018 was 991 cases, then in 2019 there were 587 cases from 33 districts. It is hoped that the police, which functions as a public safeguard, is able to respond to the phenomenon and be able to take action and uncover crimes committed against children by using an analysis of several habits that often occur simultaneously with several crimes against children. Such analysis can be performed using the Rule association technique.

The association rule is a method in data mining that looks for a set of items that often appear simultaneously (Si et al.2019), (Shaban et al. 2018), (Muhajir et al. 2020). The algorithm that is often used in the process of association rules is apriori. The Apriori algorithm performs the process of extracting information from the database in order to generate association rules (Ali et al. 2019). Problem solving in the process of extracting information from a database is done by processing the frequent itemset to generate support. Confidence. Support is the level of dominance of an item / itemset in the database, while confidence is the conditional relationship between two items (Sitnikov et al. 2018). In the case of finding patterns of crimes against children, support is used to calculate the number of each type of crime committed and confidence is used to find the relationship between the types of crimes committed over a period of time. So that the results are expected to be able to find a pattern of crime in children based on previous patterns. To generate support and confidence values, Apriori must read the database repeatedly and generate a large number of frequent itemsets and a large number of association rules. This resulted in a very high processing rate so that the achievement of support and confidence values took quite a long time to complete (Naresh et al. 2019). Apart from Apriori, there are several other algorithms for finding frequent itemsets including FP-Growth, CT-Pro, Hash-Based, Apriori Cristian Borgelt.

Dhivya and Kalpana (2010) conducted research on the performance of CT-Apriori and CT-Pro to show the speed of data execution in the form of

Siregar, A., Lydia, M. and Wage, S.
Association Rule Analysis using CT-Pro and Hash-based Algorithm in Violence Case of Children.
DOI: 10.5220/0010338800003051
In *Proceedings of the International Conference on Culture Heritage, Education, Sustainable Tourism, and Innovation Technologies (CESIT 2020)*, pages 565-573
ISBN: 978-989-758-501-2

performance curves. From the results of this study it was found that the CT-Pro was superior to the CT-Apriori algorithm by using the retail sales transaction dataset research. The CT-Pro and CT-Apriori algorithms are better than the basic algorithms, namely FP-Growth and Apriori. The difference in performance between CT-Pro and CT-Apriori is more influential at the lower threshold.

Gupta (2011) conducted a study in the form of a comparison of FP-Tree based algorithms, including COFI-Tree, CT-PRO and FP-Growth. Where FP-Growth takes a recursive approach while COFI Tree and CT-PRO take a non-recursive approach. Then in terms of FP-Growth structure, make FP-Tree, COFI Tree uses a two-way FP-Tree structure, and CT-Pro forms a Compressed FP-Tree (CFP-Tree). In terms of data execution speed, CT-PRO is better than FP Growth and COFI-Tree.

Aguru and Rao (2017) conducted research on Hash-Based using rehashing techniques with retail sales transaction research data. When the process of finding the address in the hash table occurs a collision (there is more than 1 itemset having the same hash address) and the rehashing function is used to solve the problem. At the end of their research, Aguru et al. Compared the length of execution time between Apriori and Hash-Based using the rehashing technique, the results of which the Hash-Based rehashing technique were faster than Apriori's. Hash-Based with rehashing technique with support 20 has a long execution time of 22, while Apriori with support of 20 has a long execution time of 53.

Based on the previous discussion, the CT-Pro algorithm and the Hash-Based algorithm are able to streamline the data execution time, in this case the frequent itemset search. In this study, the search for patterns of crime in children using the CT-Pro algorithm and the Hash-Based algorithm is expected to show better performance so that the achievement of support values and confidence values does not require a long time and the association rules that are formed are not too many. This study also aims to analyze the performance of the CT-Pro algorithm and the Hash-Based algorithm to search for frequent itemsets and generate association rules to get the best performance comparison of the two methods.

2 METHODS

In this study, a method to find new patterns of crime in children was developed. The CT-Pro algorithm association rules method and the Hash-Based algorithm are used by comparing the number of association rules and the length of data execution. From the prepared dataset, 150 data sets on crimes against children were obtained from the Office of Women Empowerment and Child Protection of North Sumatra Province P2TP2A unit (Integrated Service Center for Women and Children Empowerment). The data is converted into binary numbers, namely the data format in the form of 0 & 1. Each data is processed using the CT-Pro and Hash-Based algorithms. The results are used to find new patterns of crime in children, and get a comparison of the time in finding the association rules and the number of rules generated between the CT-Pro algorithm and the Hash-based algorithm.

2.1 Association Rule

Association rule is a data mining technique to identify the relationship between multiple items in a dataset (Siswanto et al., 2018). Association rules are generally of the form "if - then", with the antedecent representing "if" and "then" representing the consequent (Shaban et al., 2018). The importance of an association rule can be determined by two parameters, namely support and confidence (Segatori et al., 2018). Support is a measure or number of occurrences of items simultaneously. Confidence is a measure or percentage that states the relationship between the two items (Nomura et al., 2020).

The steps for finding association rules are divided into three stages (Ghazanfari et al. 2020).

1) Frequent itemset analysis

In this stage the process of searching for frequent itemset where the requirements are to meet or be greater than the minimum value of support (minsupport) in the database (Han et al., 2019). The support value formula as follows:

$$\text{Support} = \frac{\text{Sum of transaction X}}{\text{Total of transaction}} \text{ X } 100$$

2) The formation of association rules

Frequent itemsets are generated before the formation of association rules provided that the pattern value must be greater than the minimum confidence (minconfidence) (Ren et al., 2018). The confidence value formula as follows

$$\text{Confidence} = \frac{\text{Sum of transaction A,B}}{\text{Total of transaction A}} \text{ X } 100$$

3) The search for lift ratio

Lift ratio is a measure or unit that states whether or not an association rule is strong. The value generated from the lift ratio calculation is used to determine whether a rule is valid or not (Li et al., 2019). The size of the lift ratio is in the range of

values from 0 to infinity. (Zahrotun et al., 2018). The lift rasio value formula as follows:

$$\text{Lift Ratio} = \frac{\text{Confidence (A,B)}}{\text{Benchmark Confidence (A,B)}}$$

The benchmark confidence value using the formula:

$$\text{Benchmark Confidence} = \frac{NC}{N}$$

Notes:
Nc =Total of transactions with items as a consequent
N =Total transactions from the dataset.

2.2 CT-Pro

The flow stages of the CT-Pro algorithm include:

1) Looking for frequent itemset, where the process is selecting data against a predetermined database with the minsupport limit. Furthermore, the frequency value of each item is calculated to produce a Global item table.
2) Build a CFP-Tree, where the process is to sort frequent items in descending order based on existing Global item values and form a Global CFP-Tree.
3) Doing the frequent itemset mining process, for each item in the ordered Global item table. Search for nodes associated with these items in the Global CFP-Tree. Furthermore, local frequent items are used to build local item tables. Based on the local item table that has been formed then the Local CFP-Tree is built and frequent itemset is formed according to the items that have been mined from the Local CFP-Tree.

2.3 Hash-based

The stages of the Hash-based algorithm are:

1) Determining the minsupport value as the threshold condition for generating frequent itemset and then confidence as the threshold condition for generating the association rule.
2) C1 (Candidate 1) generation based on support calculations. Before entering each itemset into the bucket in the hash table, the hashing process for 1 itemet candidate must be done. The formula for the hashing process is
   h{x} = {order of item x} mod n.
   h = bucket address in the hash table
   n = sum of addresses, (n = 2 m + 1)
   m = total number of items
3) After performing the hash calculation, the result is C1. Itemset aims to get the hash address after calculation with the hashing formula. Itemset occupies hash addresses and becomes notes, then builds links that point to items that contain the itemset in sequence to form a link list. Then the itemset is filtered based on the minsupport value to produce L1 (Large 1).
4) The results from L1 are then combined and hashed into a hash table with the formula: H{k} = {{order of x} * 10 + order of y} mod n. If a collision occurs, it means that more than one itemset has the same hash address. The thing that must be done is rehashing with multiple addresses 2 times the previous number with the formula:
   h{k} = {{order of x} * 10 + order of y} mod j.
   Note j is the number of addresses after adding. {j = 2 * m + 1} m is the number of addresses in the hash table before adding. The addition of the hash table address is carried out until the collision between itemset is no longer found. If the result of the bucket count value is greater than or equal to the minsupport value, the L1 combination qualifies to be included in the candidate from Large itemset-2 (C2). Next is building table L2 from table C2 where the process is the same as building L1 from table C1. For searching 3-itemset use a different formula is:
   H(k) = ((order of X) * 100 + (order of Y) * 10 + order of Z) mod j.
   Order of Z states the order of items from the third item.

3 RESULT AND DISCUSSION

3.1 CT-Pro

In this study, 150 datasets in the form of crime data on children from the Office of Women's Empowerment and Child Protection of North Sumatra Province, P2TP2A unit were used. The data was converted in the form of binary numbers, namely the data format is in the form of 1 & 0. The value is 1 if there is a crime criterion in the case and a value of 0 if there is no crime criterion in the case. For example, in the first case there were crimes PF, PE, PN and TR.

Table 1: Data Conversion.

NO	PF	PS	PE	PP	PN	TR	MA	PB	EP
1	1	0	1	0	1	1	0	0	0
2	1	0	1	0	0	1	0	0	0
3	1	1	1	0	0	0	0	0	0
4	0	1	1	0	0	0	0	1	0
5	1	0	1	1	0	0	0	0	0
6	0	1	1	0	0	1	0	0	0
7	1	0	1	1	0	0	1	0	0
8	0	0	1	1	1	0	0	0	0
9	1	0	0	0	0	0	0	0	0
10	0	1	0	0	0	1	0	0	0
11	1	0	1	1	0	0	0	0	0
12	1	0	1	0	0	0	0	1	0
13	1	1	0	0	0	0	0	0	0
14	1	1	1	0	0	1	0	0	0
15	1	1	1	0	0	0	0	0	0
16	0	0	0	1	0	0	1	0	0
17	1	0	1	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	1
19	0	1	1	0	0	1	0	0	0
20	0	1	1	0	1	0	0	0	0

Notes:
Physical Torture = PF, Sexual harassment = PS, Emotional Torture = PE, Abandonment and Neglect = PP, Rejection = PN, Giving Terror to Children = TR, Isolating Children = MA, Giving Bad Influence to Children = PB, Exploitation = EP.

The next step was to create a Global item table where each item was filtered with a predetermined minsupport value of 10%. Furthermore, the data were sorted from the largest to the smallest frequency (descending) until a global item table is formed. The PE itemset with the largest support count, namely 15, get global ID 1.And PB itemset with the smallest support, namely 2, with global ID 8.

Table 2: Global item.

Global ID	Itemset	Support
1	PE	15
2	PF	12
3	PS	9
4	TR	6
5	PP	5
6	PN	3
7	MA	2
8	PB	2

Then perform data mapping, mapping is data mapping against the global ID table in table 2. In the first case there was cases of PF, PE, PN and TR where the global IDs of the cases were 1, 2, 4 and 6. The next step is to build a Global CFP-Tree by following the following processes. (i) Forming a new node for each item in the global item table; (ii) Accessing each item in the itemset, if the item in the itemset is currentNode, then the number in the current node is added by one, but if the item is not the same as currentNode, a new node will be created for the item. (iii) Each time the process of creating a new node, setting the next and prev attribute values is done; (iv) The process continues until all items are accessed.

After the Global CFP-Tree is formed, the mining process was carried out. In carrying out the Global item table mining process, data was sorted based on data from the smallest to the largest frequencies. At this stage, take the PS (Sexual Harassment) data for example with a support count of 9, the sixth smallest data based on the global item table. The next step was to find nodes that have links to PS in the Global CFP-Tree, hereinafter referred to as Local frequent items and used to build a Local item table then a Local CFP-Tree was built as shown in Figure 1:

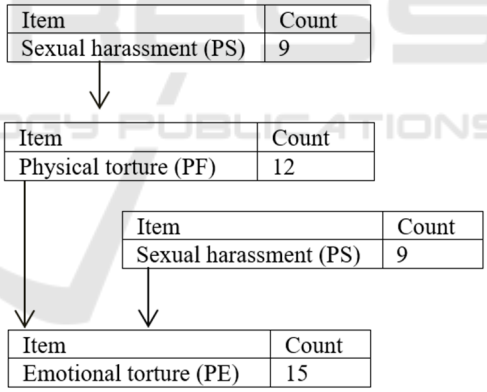

Figure 1: Local CFP-Tree.

Then from the Local CFP-Tree, the PS frequent itemset was obtained:

- Sexual harassment (PS).
- Physical Torture(PF) - Sexual harassment(PS), Emotional torture (PE) - Physical torture (PF), Emotional torture (PE) - Sexual Harassment (PS).
- Emotional torture (PE) - Physical Torture (PF)-Sexual harassment (PS).

Based on the frequent itemset, the confidence value with a minconfidence ≥ 60% was calculated. For example, from the frequent itemset (PS-PF-PE) to search for the combination and calculate the

confidence value. The following is the calculation result of the confidence value for several itemset:

1. Confidence (Sexual harassment => Physical Torture).
$$= \frac{\sum \text{Sexual harassment and Physical torture}}{\sum \text{Sexual harassment}}$$
= 4 / 9 = 0.44 * 100 % = 44 %

2. Confidence (Physical Torture => Sexual harassment).
$$= \frac{\sum \text{Physical torture and Sexual harassment}}{\sum \text{Physical torture}}$$
= 4 / 12 = 0.33 * 100 % = 33 %

3. Confidence (Sexual harassment => Emotional torture).
$$= \frac{\sum \text{Sexual harassment and Emotional torture}}{\sum \text{Sexual harassment}}$$
= 7 / 9 = 0.77 * 100 % = 77 %

4. Confidence (Emotional torture => Sexual harassment.
$$= \frac{\sum \text{Emotional torture and Sexual harassment}}{\sum \text{Emotional torture}}$$
= 7 / 15 = 0.46 * 100 % = 46 %

5. Confidence (Physical Torture => Emotional torture).
$$= \frac{\sum \text{Physical torture and Emotional torture}}{\sum \text{Physical torture}}$$
= 10 / 12 = 0.83 * 100 % = 83 %

6. Confidence (Emotional torture => Physical Torture).
$$= \frac{\sum \text{Emotional torture and Physical torture}}{\sum \text{Emotional torture}}$$
= 10 / 15 = 0.66 * 100 % = 66 %

7. Confidence (Sexual harassment => Physical Torture => Emotional torture).
$$\frac{=\sum \text{Sexual harassment ,Physical torture and Emotional torture}}{\sum \text{Sexual harassment}}$$
= 3 / 9 = 0.33 * 100 % = 33 %

8. Confidence (Physical Torture => Emotional torture => Sexual harassment).
$$\frac{=\sum \text{Physical torture ,Emotional torture,Sexual harassment}}{\sum \text{Physical torture}}$$
= 3 / 12 = 0.25 * 100 % = 25 %

9. Confidence (Emotional torture => Sexual harassment => Physical Torture).
$$\frac{=\sum \text{Emotional torture=> Sexual harassment => Physical torture}}{\sum \text{Emotional torture}}$$
3 / 15 = 0.2 * 100 % = 20 %

After obtaining a rule that meets the minimum confidence which the rule has a minsupport> 10% and a minconfidence> 60%. The result is that there were 3 itemsets that meet these rules, namely PE-PF, PS-PE, PF-PE. Furthermore, benchmark confidence (BC) was calculated to obtain the lift ratio value. Where the benchmark confidence was generated by dividing the number of consequent occurrences (Nc) then divided by the number of data (N). From these results, the lift ratio was then searched by dividing the value of confidence and benchmark confidence. The result, if an act of emotional abuse is committed then there is no crime of physical torture. Confidence: 66%, Support count: 10 and Lift Ratio: 1.1, if a crime of sexual harassment is committed then a crime of emotional torture will occur. Confidence: 77%, Support count: 7 and Lift Ratio: 1.02, if the crime of physical torture is committed then there will be no crime of emotional torture. Confidence: 83%, Support count: 10 and Lift Ratio: 1.10. From the calculation results obtained in the lift ratio value table obtained and successfully formed which has a value greater than one (lift ratio> 1) indicates that the rule is strong and valid. And vice versa if (lift ratio <1), it indicates that the rule is not strong or invalid.

3.2 Hash-based

Hash-based processes were tested using the same data as many as 150 datasets in the form of child crime data. The stage of the hash-based algorithm is to determine the value of minsupport and minconfidence as a threshold condition, minsupport >10% and minconfidence >60%. To simplify the calculation of the hash table, each item requires a sequence of items in the data which is used to represent the values in the calculation. For example the Emotional Torture itemset with Initial PE in the order of 1, following is the order of the items that have been determined in Table 3.

Table 3: Order of item.

Initial	Itemset	Order
PE	Emotional Torture	1
PF	Physical Torture	2
PS	Sexual Harassment	3
TR	Giving Terror to Children	4
PP	Abandonment and Neglect	5
PN	Rejection	6
MA	Isolating Children	7
PB	Giving Bad Influence to Children	8
EK	Exploitation	9

The generation of C1 was carried out based on the calculation of support count. Before entering each itemset into the bucket in the hash table, the hashing process for the 1-itemset candidate must be done with

the formula h{x} = {order of item x} mod n. Address lookup in the hash table for 1 itemset:

h (Emotional Torture) = (1) mod 19 = 1
h (Physical Torture) = (2) mod 19 = 2
h (Sexual Harassment) = (3) mod 19 = 3
h (Giving Terror to Children) = (4) mod 19 = 4
h(Abandonment and Neglect) = (5) mod 19 = 5
h (Rejection) = (6) mod 19 = 6
h (Exile Children) = (7) mod 19 = 7
h (Bad Influence) = (8) mod 19 = 8
h (Exploitation) = (9) mod 19 = 9

After performing the hash calculation, the itemset gets the hash address. Itemset occupies hash addresses and becomes notes, then builds links that point to items that contain the itemset sequentially until the link list is formed. Then the itemset was filtered based on the minsupport value, which is >10%, itemset that has a support value> 10% will produce L1 (Large 1). The result of the itemset with the highest support was PE, which is 15 Count with index 1 and the lowest support itemset was PB, which is 2 Count with index 8. Itemset Large 1 is shown in Table 4.

Table 4: L1 (Large 1).

Index	Itemset	Support
1	PE	15
2	PF	12
3	PS	9
4	TR	6
5	PP	5
6	PN	3
7	MA	2
8	PB	2

The large 1 table is data sorted from the largest to the smallest frequency (descending) after going through the selection process at C1 (Candidate 1). The results from L1 are then combined and hashed into the hash table with the formula: H {k} = {{order of x} * 10 + order of y} mod n.

Address lookup in hash table for 2-itemset:

h (PE, PF) = ((1) * 10 + 2) mod 19 = 12
h (PE, PS) = ((1) * 10 + 3) mod 19 = 13
h (PE, TR) = ((1) * 10 + 4) mod 19 = 14
h (PE, PP) = ((1) * 10 + 5) mod 19 = 15 *
h (PE, PN) = ((1) * 10 + 6) mod 19 = 16
h (PE, MA) = ((1) * 10 + 7) mod 19 = 17*
h (PE, PB) = ((1) * 10 + 8) mod 19 = 18*
h (PF, PS) = ((2) * 10 + 3) mod 19 = 4
h (PF, TR) = ((2) * 10 + 4) mod 19 = 5
h (PF, PP) = ((2) * 10 + 5) mod 19 = 6
h (PF, PN) = ((2) * 10 + 6) mod 19 = 7
h (PF, MA) = ((2) * 10 + 7) mod 19 = 8*
h (PF, PB) = ((2) * 10 + 8) mod 19 = 9
h (PS, TR) = ((3) * 10 + 4) mod 19 = 15*
h (PS, PN) = ((3) * 10 + 6) mod 19 = 17 *
h (PS, PB) = ((3) * 10 + 8) mod 19 = 0*
h (TR, PN) = ((4) * 10 + 6) mod 19 = 8*
h (PP, PN) = ((5) * 10 + 6) mod 19 = 18*
h (PP, MA) = ((5) * 10 + 7) mod 19 = 0 *

In the calculation above, a collision is found, which means there is more than one itemset that has the same hash address. In this calculation, the collision is at the 0 address (PS, PB) with (PP, MA), the 8th address (PF, MA) with (TR, PN), the 15th address (PE, PP) with (PS, TR), and the 17th address (PE, MA) with (PS, PN), the 18th address (PE, PB) with (PP, PN). If a collision occurs, the first thing to do is check or check the available bucket address. If after checking is done and an indication is found that the hash table has been filled, then rehashing with multiple addresses 2 times the number of previous addresses must be done with the formula:

h {k} = {{order of x} * 10 + order of y} mod j,
j is the number of addresses after adding. {j = 2 * m + 1} m is the number of addresses in the hash table before adding.

h (PE, PF) = ((1) * 10 + 2) mod 39 = 12
h (PE, PS) = ((1) * 10 + 3) mod 39 = 13
h (PE, TR) = ((1) * 10 + 4) mod 39 = 14
h (PE, PP) = ((1) * 10 + 5) mod 39 = 15
h (PE, PN) = ((1) * 10 + 6) mod 39 = 16
h (PE, MA) = ((1) * 10 + 7) mod 39 = 17*
h (PE, PB) = ((1) * 10 + 8) mod 39 = 18*
h (PF, PS) = ((2) * 10 + 3) mod 39 = 23
h (PF, TR) = ((2) * 10 + 4) mod 39 = 24
h (PF, PP) = ((2) * 10 + 5) mod 39 = 25
h(PF, PN) = ((2) * 10 + 6) mod 39 = 26
h (PF, MA) = ((2) * 10 + 7) mod 39 = 27
h (PF, PB) = ((2) * 10 + 8) mod 39 = 28
h (PS, TR) = ((3) * 10 + 4) mod 39 = 34
h (PS, PN) = ((3) * 10 + 6) mod 39 = 36
h (PS, PB) = ((3) * 10 + 8) mod 39 = 38
h (TR, PN) = ((4) * 10 + 6) mod 39 = 7
h (PP, PN) = ((5) * 10 + 6) mod 39 = 17*
h (PP, MA) = ((5) * 10 + 7) mod 39 = 18*

It was also found that collisions at the 17th address for (PP, PN) with (PE, MA) and the 18th address for (PP, MA) with (PE, PB) still occurred. To solve this problem, the same formula is used again.

h (PE, PF) = ((1) * 10 + 2) mod 79 = 12
h (PE, PS) = ((1) * 10 + 3) mod 79 = 13

h (PE, TR) = ((1) * 10 + 4) mod 79 = 14
h (PE, PP) = ((1) * 10 + 5) mod 79 = 15
h (PE, PN) = ((1) * 10 + 6) mod 79 = 16
h (PE, MA) = ((1) * 10 + 7) mod 79 = 17
h (PE, PB) = ((1) * 10 + 8) mod 79 = 18
h (PF, PS) = ((2) * 10 + 3) mod 79 = 23
h (PF, TR) = ((2) * 10 + 4) mod 79 = 24
h (PF, PP) = ((2) * 10 + 5) mod 79 = 25
h (PF, PN) = ((2) * 10 + 6) mod 79 = 26
h (PF, MA) = ((2) * 10 + 7) mod 79 = 27
h (PF, PB) = ((2) * 10 + 8) mod 79 = 28
h (PS, TR) = ((3) * 10 + 4) mod 79 = 34
h (PS, PN) = ((3) * 10 + 6) mod 79 = 36
h (PS, PB) = ((3) * 10 + 8) mod 79 = 38
h (TR, PN) = ((4) * 10 + 6) mod 79 = 46
h (PP, PN) = ((5) * 10 + 6) mod 79 = 56
h (PP, MA) = ((5) * 10 + 7) mod 79 = 57

The addition of the hash table address is carried out until the collision between itemset is no longer found. Each address is filled with 1 itemset then the combined L1 (L1 * L1) results are then distributed into the address bucket. From the hash table, the calculation of support for frequent 2-itemset using the support formula is performed. The results show that the PE, PF itemset with address 12 has a support percentage of 50% and a support count of 10 from the total data of 20 cases. The complete calculation result of frequent 2-itemset or C2 can be seen in Table 5.

Table 5: Frequent 2-Itemset (Tabel C2).

Address	Itemset	Count	N	Support
12	(PE, PF)	10	20	50 %
13	(PE, PS)	7	20	35 %
14	(PE, TR)	5	20	25 %
15	(PE, PP)	4	20	20 %
16	(PE, PN)	3	20	15 %
17	(PE, MA)	1	20	5 %
18	(PE, PB)	2	20	10 %
23	(PF, PS)	4	20	20 %
24	(PF, TR)	4	20	20 %
25	(PF, PP)	3	20	15 %
26	(PF, PN)	1	20	5 %
27	(PF, MA)	1	20	5 %
28	(PF, PB)	1	20	5 %
34	(PS, TR)	5	20	25 %
36	(PS, PN)	1	20	5 %
38	(PS, PB)	1	20	5 %
46	(TR, PN)	1	20	5 %
56	(PP, PN)	1	20	5 %
57	(PP, MA)	2	20	10 %

From Table 6, the itemset which has a minsupport value of >10% is then carried out to produce frequent 2-itemset or L2. Followed by looking for the confidence formula value as follows:

$$\text{Confidence} = \frac{\text{Sum of transaction A and B}}{\text{Total Transaction A}} \text{X } 100$$

If the minconfidence value is > 60% then the value below the minconfidence will be eliminated. From this calculation, there is one itemset that has a value of> 60% itemset, namely PE, PF with a number of support counts A and B of 10 and support count A of 15. Then proceed with the calculation of benchmark confidence and lift ratio to find out whether the rule is valid or not. . Based on the calculations carried out, it can be concluded that those who meet minutes support> 10%, minimum confidence> 60% and lift ratio> 1 are as follows: If an act of emotional torture (PE) is committed then there will be no crime of physical torture (PF). Confidence: 66%, Support Count: 10 and Lift Rasio 1.1.

Next, to look for frequent 3-itemset, L2 results are combined and hashed into a hash table with the formula: H(k) = ((order of X) * 100 + (order of Y) * 10 + order of Z) mod j.

Based on the first test conducted with data from 150 cases, the CT-Pro algorithm obtained minsupport = 15% and minconfidence = 50% with 2 rules generated by the number of rules, and 0.06 seconds execution time. Meanwhile, Hash-Based generates 2 rules, with an execution time of 0.41 seconds. The second test was carried out with the CT-Pro algorithm with minsupport = 10% and minconfidence = 40% with the number of rules generated as many as 8 rules and an execution time of 0.07 seconds. Meanwhile, Hash-Based generates 8 rules, with an execution time of 0.43 seconds.

The following are the complete results of the comparison test between the CT-Pro algorithm and the Hash-Based algorithm:

Table 6: Comparison Results.

No	Min supp %	Min conf %	CT-Pro		Hash-Based	
				Time	Rule	Time (sec)
1	15	50	2	0.06	2	0.41
2	10	40	8	0.07	8	0.43
3	7	30	13	0.11	13	0.48
4	5	20	20	0.16	20	0.58
5	3	15	22	0.25	22	0.73

Execution time comparison chart:

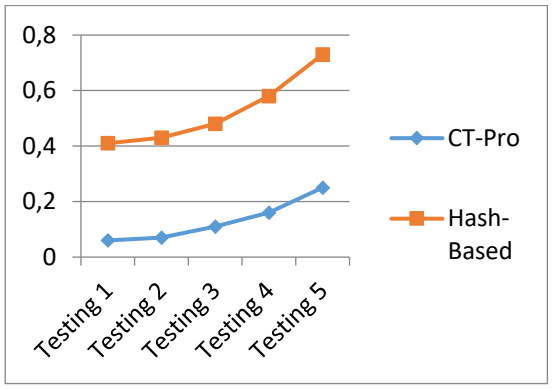

Figure 2: Execution Time Comparison Result.

The results of the conducted tests shows that the smaller of the given minsupport and the minconfidence values, the longer the data execution time will be (since more association rules were formed). Conversely, the higher the given minutes support and the minconfidence values, the faster the data execution time will be (since fewer association rules were formed). In this study, the CT-Pro algorithm was proven to work well. This can be seen from the CFP-Tree data structure where the number of nodes built was very limited so that data execution was faster. Meanwhile, the Hash-Based algorithm selects data in the generation process C1 (candidate 1) and L1 (Large 1) and so on, using the hashing formula. In the hashing calculation process, each item must have a different address. If there is the same address (collision), then re-hashing is done by adding the number of addresses, which is 2 times the previous number plus 1. In the calculation of the dataset above, there were several collisions so that there was an addition of the address. This causes the Hash-Bases process to take a long time to execute data.

4 CONCLUSION

From the comparison test results between the CT-Pro algorithm and the Hash-Based algorithm, it can be concluded that the CT-Pro algorithm produces a faster or better processing time than the Hash-Based algorithm. The conducted test results shows that a minimum support and confidence of 3% and 15%, respectively, and CT-Pro produces 22 rules with an execution time of 0.25 seconds were obtained. The result is faster than the Hash-Based algorithm which generates 22 rules with an execution time of 0.73 seconds. This difference occurs due to collisions which cause an increase in the number of addresses in the hashing process. A new crime pattern with the highest support and confidence was found if there was an act of sexual harassment where there would be physical torture with a confidence of 59%, a support count of 34 and a lift ratio of 1.29.

REFERENCES

Aguru, S. and Rao, B.M., 2014. A Hash Based Frequent Itemset Mining using Rehashing, *International Journal on Recent and Innovation Trends in Computing and Communication, Volume: 2 Issue: 12.*

Ali, Y., Farooq, A., Alam, T. M., Farooq, M. S., Awan, M. J., & Baig, T. I., 2019. Detection of Schistosomiasis Factors using Association Rule Mining, *IEEE Access*, 2019.2956020:1-2019.2956020:8.

Atmaja, E.H.S., Simaremare, R. and Rosa, P.H.P., 2019. Aplication of CT-Pro Algorithm For Crime Analysis, *Conference SENATIK STT Adisutjipto Yogyakarta*. pp.435-444.

Dhivya, A.B. and Kalpana, B., 2010. A study on the Performance of CT-APRIORI and CT-PRO Algorithms using Compressed Structures for Pattern Mining, *Journal of Global Research in Computer Science*, 1(2), pp. 8-15.

Ghazanfari, B., Afghah, F. and Taylor, M.E., 2020. Sequential Association Rule Mining for Autonomously Extracting Hierarchical Task Structures in Reinforcement Learning, *IEEE ACCESS* 2020:2965930:1-2020:2965930:18.

Gupta, B. and Garg, D., 2011. FP-Tree Based Algorithms Analysis: FP-Growth, COFI-Tree and CT-PRO, *International Journal on Computer Science and Engineering (IJCSE)*. 3(7) pp. 2691-2699.

Han, Q., Lu, D., Zhang, K., Song, H., & Zhang, H., 2019. Secure Mining Of Association Rules In Distributed Datasets, *IEEE Access*. 2019:2948033:1-2019:2948033:10. 2019.

Hossain, M., Sattar A.H.M. and Paul, M.K., 2019. Market Basket Analysis Using Apriori and FP Growth Algorithm, *International Conference on Computer and Information Technology (ICCIT).*

Islamiyah., Ginting, P.L., Dengen, N. and Taruk, M., 2019. Comparison of Apriori and FP-Growth Algorithms in Determining Association Rule, *International Conference on Electrical, Electronics and Information Engineering (ICEEIE 2019)*. pp.320-323.

Law of the Republic of Indonesia No. 23. 2002. *Concerning Child Protection.* State institutions Republic of Indonesia. 2002; 109: 1-14.

Li, A., Liu, L., Ullah, A., Wang, R., Ma, J., Huang, R., Yu, H., Ning, H., 2019. Association Rule-Based Breast Cancer Prevention and Control System, *IEEE Transactions on Computational Social Systems. Pp.1106-1114.*

Muhajir, M., Kusumawati, A. and Mulyadi, S., 2020. Apriori Algorithm for Frequent Pattern Mining for Public Librariesin United States, *Proceedings of the*

International Conference on Mathematics and Islam (ICMIs 2018). pp.60-64

Naresh, P. and Suguna, R., 2019. Association Rule Mining Algorithms on Large and Small Datasets: A Comparative Study, *Proceedings of the International Conference on Intelligent Computing and Control Systems (ICICCS 2019).* pp.587-592

Nomura, K., Shiraishi, Y., Mohri, M. and Morii, M., 2020. Secure Association Rule Mining on Vertically Partitioned Data Using Private-Set Intersection, *IEEE Access.* 2020:3014330:1-2020:3014330:10

Rao, S. and Gupta, P., 2012. Implementing Improved Algorithm Over Apriori Data Mining Association Rule Algorithm, *IJCST*. Vol. 3. pp. 489-493. Jan-Mar 2012.

Ren, F., Pei, Z., & Wu, K., 2019. Selection of Satisfied Association Rules via Aggregation of Linguistic Satisfied Degrees, *IEEE Access.* 2019:2926735:1-2019:2926735:17. 2019

Segatori, A., Bechini, A., Ducange, P. and Marcelloni, F., 2018. A Distributed Fuzzy Associative Classifier for Big Data, *IEEE Transactions on Cybernetics*. pp.2656–2669

Shaban, A., Almasalha, F., & Qutqut, M. H.,2018. Hybrid user action prediction system for automated home using association rules and ontology, *IET Wireless Sensor Systems.* Vol. 9 Iss. 2. pp. 85-93

Si, H., Zhou, J., Chen, Z., Wan, J., Xiong, N.N., Zhang, W., 2019. *Association Rules Mining among Interests and Applications for Users on Social Networks*, IEEE Access 2019:2925819:1-2019:2925819:13

Siswanto, B., Thariqa, P., 2018. Association Rules Mining for Identifying Popular Ingredients on YouTube Cooking Recipes Videos, *INAPR International Conference*. pp. 95-98.

Sitnikov, D., Titova, O., Minukhin, S., Kovalenko A., and Titov S., 2018. Informativity of Association Rules from the Viewpoint of Information Theory, *International Scientific-Practical Conference.* pp.595-598 .

Zahrotun, L., Soyusiawaty, D. and Pattihua, R.S., 2018. The Implementation of Data Mining for Association Patterns Determination Using Temporal Association Methods in Medicine Data*, Internasional Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*. pp.668-673.