khazanah
informatika

Khazanah Informatika: Jurnal Ilmu Komputer dan Informatika is a national scientific journal that publishes scientific research papers/articles or reviews in the field of Computer Systems and Informatics. The journal is accredited “Sinta 2” according to the decree of Directorate of DPRP (Dirjen Penguatan Riset dan Pengembangan), Ministry of Higher Education number 200/M/KPT/2020 dated December 30, 2020. The decree is valid from vol. 6 issue 2 (2020) until vol. 11 no. 2 (2025). The scope of this journal includes software engineering, information systems development, computer systems, and computer networking. The journal is published by Muhammadiyah University Press (MUP), Universitas Muhammadiyah Surakarta.

ISSN: 2621-038X e-ISSN: 2477-698X

khazanah informatika

Vol. 7, No. 2 | October 2021

Table of Contents

khazanah informatika

Jurnal Ilmu Komputer dan Informatika

Complex University Timetabling Using Iterative Forward Search Algorithm and Great Deluge Algorithm

I Gusti Agung Premananda*, Ahmad Muklason
Information Systems Department
Faculty of Intelligent Electrical and Informatics Technology
Institut Teknologi Sepuluh Nopember Surabaya
Surabaya, Indonesia
*igustiagungpremananda@gmail.com

Abstract-University timetabling is an issue that has received more attention in the field of operations research. Course scheduling is the process of arranging time slots and room for a class by paying attention to existing limitations. This problem is an NP-Hard problem, which means the computation time to find a solution increases exponentially with the size of the problem. Solutions to problems of this kind generally use a heuristic approach, which tries to find a sufficiently good (not necessarily optimal) solution in a reasonable time. We go through two stages in solving the timetabling problem. The first stage is to schedule all classes without breaking any predefined rules. The second stage optimizes the timetable generated in the first stage. This study attempts to solve the class timetabling problem issued in a competition called the 2019 International Timetabling Competition (ITC 2019). In the first stage, we use the Iterative Forward Search (IFS) algorithm to eliminate timetable candidates and to generate a schedule. In the second stage, we employ the Great Deluge algorithm with a hyper-heuristic approach to optimize the solution produced in the first stage. We have tested the method using 30 datasets by taking 1,000,000 iterations on each dataset. The result is an application that does schedule elimination and uses the IFS algorithm to produce a schedule that does not violate any of the hard constraints on 30 ITC 2019 datasets. The implementation of the Great Deluge algorithm optimizes existing schedules with an average penalty reduction of 42%.

Keywords: timetabling, class scheduling, iterated forward search, international timetabling competition

Article info: *submitted December 10, 2020, revised March 4, 2021, accepted March 24, 2021*

1. Introduction

Timetabling problems in education have received much attention and have long been studied in the field of operations research [1]. This problem contains how to schedule courses against the available schedule and room. There are two types of constraints pertaining to the challenge, namely hard constraints and soft constraints [2]. Hard constraint is a limit that must be met in scheduling a class [3]. Examples of this limitation include the maximum capacity of a classroom, the number of schedule slots available in a class and two classes that cannot be scheduled simultaneously. Soft constraint is a limit that can be violated, but if it is violated, it will cause the quality of scheduling to decrease [3]. Generally, the penalty value is used on the soft constraints that are violated to measure the quality of the resulting scheduling. Examples of soft constraints such as preferably class a and class b should be scheduled simultaneously, class a should not be scheduled in room x, and class a should be scheduled in a different week from class c.

Scheduling problems involve many rooms and many possible time slots. Hence the number of possible combinations will be very high. This problem has been categorized as an NP-Hard problem [4] which means the computation time required to find a solution increases exponentially with the size of the problem [5]. If there are 3 classes to be scheduled, the number of possible schedule combinations is 6. However, if there are 10 classes to be scheduled, the number of possible schedule combinations increases exponentially to 3,628,800 possibilities. Facing this fact, latest researches focus more on developing algorithms with a heuristic approach, namely an approach to produce a fairly good solution (not necessarily the optimal one) and within a reasonable time (time that allows it to be applied to real problems) [6].

Generally solving scheduling problems consists of two phases, namely the first phase to build an initial solution without breaking existing hard constraints and the next phase is to optimize the solution by reducing the number of penalties for violating soft constraints [7]. In the first phase, Graph Coloring algorithm is generally applied, which is a simple algorithm to convert classes into graphical form and use coloring to ensure that there are no violations of hard constraints [8]. However, in several case studies, sometimes there are too many hard constraints that the Graph Coloring algorithm could not produce an initial solution. This calls for another algorithm to tackle the challenge [7].

Several studies have been conducted to solve timetabling problems. Research conducted by Muller et al in 2004 [9] created a new algorithm that can be used in university timetabling problems in which there are many constraints and hierarchies of courses so that this problem is known as *complex university timetabling* [10]. This algorithm is called Iterative Forward Search (IFS). The trial was carried out on scheduling problems at Purdue University and was able to meet up to 98% of student requests where there were no conflicting schedules between the courses selected by the student. Subsequent research was carried out by Rudová in 2010 [10] where they successfully applied the IFS algorithm to solve scheduling problems in the 2007 ITC dataset. Meanwhile, on the same dataset, Muklason conducted research in 2019 [11] to perform optimization using the Great Deluge algorithm based on hyper heuristics scheme. The result concluded that this algorithm is superior to other algorithms such as simulated annealing and hill climbing algorithms [11].

Table 1. Description of the ITC 2019 Dataset

Dataset	Number of Classes	Number of Rooms	Number of students	Number of Limits
agh-fis-spr17	1239	80	1641	1220
agh-ggis-spr17	1859	44	2116	2690
bet-fal17	983	62	3018	1251
iku-fal17	2641	214	0	2902
mary-spr17	882	90	3666	3947
muni-fi-spr16	575	35	1543	740
muni-fsps-spr17	561	44	865	400
muni-pdf-spr16c	2526	70	2938	2026
pu-llr-spr17	1001	75	27018	634
tg-fal17	711	15	0	501
agh-ggos-spr17	1144	84	2254	1688
agh-h-spr17	460	39	1988	399
lums-spr18	487	73	0	518
muni-fi-spr17	516	35	1469	699
muni-fsps-spr17c	650	29	395	709
muni-pdf-spr16	1515	83	3443	1012
nbi-spr18	782	67	2293	596
pu-d5-spr17	1061	84	13497	1535
pu-proj-fal19	8813	768	38437	7797
yach-fal17	417	28	821	645
agh-fal17	5081	327	6925	7154
bet-spr18	1083	63	2921	1418
iku-spr18	2782	208	0	3488
lums-fal17	502	73	0	597
mary-fal18	951	93	5051	513
muni-fi-fal17	535	36	1685	787
muni-fspsx-fal17	1623	33	1152	1359
muni-pdfx-fal17	3717	86	5651	3501
pu-d9-fal19	2798	224	35213	2746
tg-spr18	676	18	0	426

The 2019 ITC [12] issued 30 new datasets (Table 1) containing timetabling problems based on original data from several universities in the world. This problem contains how to schedule classes without violating existing restrictions and also produce good quality timetable by obtaining the smallest possible penalty. The penalty value at ITC 2019 comes from four types of soft constraints. The first is that each schedule candidate has a different penalty value. So that as much as possible in scheduling a class, it is done on the schedule candidate having the lowest penalty. Furthermore, each room also has a different penalty value so that as much as possible we choose the room with the lowest penalty when scheduling a class. Furthermore, a penalty is applied if a student gets two or more overlapped classes – hence impossible to attend them all. Penalties will be given according to the number of overlapping classes for each student. Finally, penalties are awarded based on violations of 19 types of distribution (Table 2) which are soft constraints.

Table 2. Description of Distribution Boundaries

Distribution Limits	Description
SameStart	The classes covered by this limit must start in the same time slot
SameTime	Classes that fall under this limit must have the same start and end times if the class duration is the same. If the class duration is different, the shorter class must start at the same time or after the class whose duration is longer and must end before or together with the longer duration class.

Distribution Limits	Description
DifferentTime	The opposite of the sameTime limitation.
SameDays	Classes that are subject to this limit must be scheduled on the same day. If a class has fewer days in the week, that class should be scheduled as a subset of the class with the larger number of days.
DifferentDays	The opposite of the SameDays limitation
SameWeeks	These limits are the same as the SameDays limits, but apply on a week basis
DifferentWeeks	The opposite of the SameWeeks constraint
Overlap	Classes that fall under this limitation must be scheduled overlapping in terms of time slots, days and weeks.
NotOverlap	The opposite of the Overlap limit
SameRoom	Classes that fall under this limit must be scheduled in the same room
DifferentRoom	The opposite of SameRoom limitation
SameAttendees	The class that is in this limit must allow one student to take it at the same time. Classes must not be scheduled overlap and must pay attention to the distance to go from one class to another.
Precedence	Classes contained in this constraint must be scheduled according to the order in this constraint. The ordering is only based on the first time meeting in each class.
WorkDay	Classes that are subject to this limit may not be scheduled for the end of the final class plus the initial class start over the S time slot if it is scheduled on the same day and week.
MinGap	Classes that fall under this limit must have a certain distance (x) of time if scheduled on the same day.
MaxDays	Classes that are subject to this limit cannot be scheduled more than (x) days apart.
MaxDayLoad	Classes subject to this limit must be scheduled for no more than (x) time slots on each day.
MaxBreaks	Classes that fall under this limit must be scheduled with no break between classes more than (x) time slots on each day.
MaxBlock	This limitation imposes a limit on the block length (some classes are scheduled with the rest interval between classes is less than (y) time slots) so that there are no more than (x) time slots on each day.

This dataset has a higher level of complexity compared to the dataset issued by the same competition in 2011, 2007 and 2002. The increased complexity lies in the limited list of schedules in each class, increased types of hard constraints and soft constraints to 19 types, and the presence of hierarchies in each subject. The ITC 2019 dataset is divided into 3 types, namely 10 early instance datasets, 10 middle instance datasets and 10 late instance datasets. The three groups of the dataset have different levels of difficulty based on the number of classes, the number of rooms, the number of students and the number of distributions. The *early* dataset is the easiest dataset with the smallest average number of classes, number of rooms, number of students, and distribution. Furthermore, the *middle* dataset has a moderate level of difficulty with the average number of classes, the number of rooms, the number of students and distribution more than the *early* group dataset but less than the *late* group dataset. The *late* dataset group is the most difficult dataset with an average number of classes, the number of rooms, the number of students and the most distribution compared to the *early* and *middle* group dataset.

Based on the above background, this study aims to solve the latest complex scheduling problems using the latest dataset from ITC 2019. From this research it is hoped that it can be used by various universities that have similar problems in producing course schedules in their respective departments or universities.

2. Methods

This section explains the stages carried out in this study, starting from data preprocessing to validating the final solution.

a. Data Preprocessing

At this stage there will be elimination of several schedule candidates in each class that are impossible to use. This happens because the candidate's schedule and room have clearly violated the hard constraints. Otherwise, if those candidates were retained, it will interfere with the initial solution search process and the solution optimization.

The first elimination is carried out based on room usage restrictions, where there are rooms that cannot be used at a certain time. Each candidate schedule will be checked as shown in Figure 1. If there is a conflict, the candidate schedule will be deleted.

The next elimination is carried out on schedule candidates who violate the distribution constraints (Table 2) in the form of hard constraints that present in each class. For example if there is a distribution limitation where class (x) must be scheduled on the same day (SameDay) as class (y), then the candidate schedule in class (x) which has the same day as the class schedule candidate (y) will be retained and the rest going to elimination. All schedule candidates in each class will be checked against the distribution boundaries in the form of hard constraints as shown in Figure 2. If it is found that there is a violation of any hard constraint distribution, the candidate schedule will be discarded.

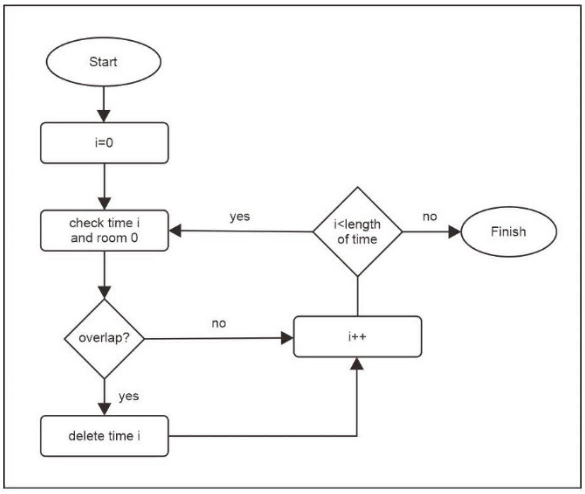


Figure 1. First Elimination Flow

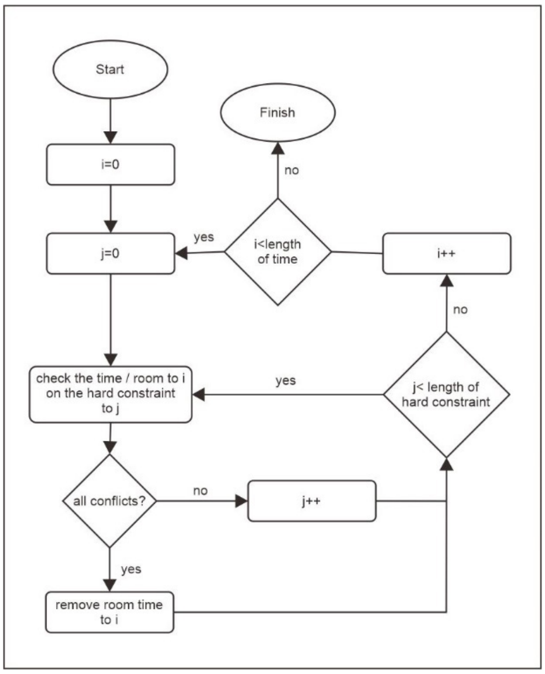


Figure 2. Second Elimination Flow

The next elimination is carried out based on the class which has one schedule candidate. For example, suppose class (x) has only one schedule candidate. Then class (x) will be scheduled on that one candidate schedule because there is no other choice. Apart from class (x), if there is a class that has an overlap with the candidate schedule in class (x), it will be eliminated because it is not possible to use it. The flow of this elimination can be seen in Figure 3. To see how effective this elimination is, the results will be compared with elimination and without elimination.

b. Algorithm design and implementation

At this stage, a design will be carried out to make several changes to the IFS algorithm so that it is able to produce a viable solution. The IFS algorithm initially takes a variable randomly and tries to enter a randomly drawn value from that variable. If there is a conflict, the value in the conflict variable will be temporarily deleted. The pseudo code of the IFS algorithm can be seen in Figure 4.

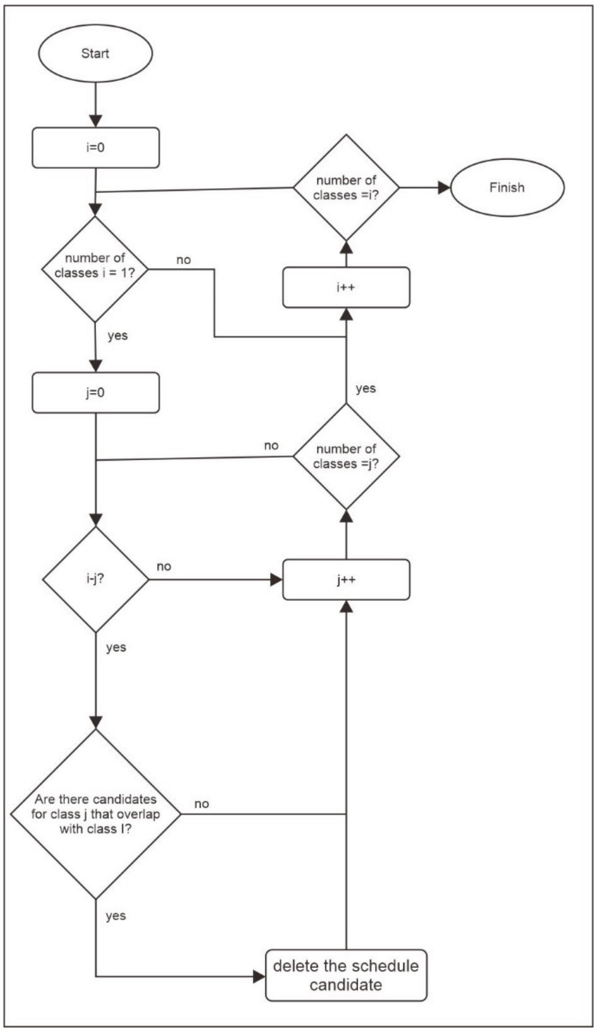


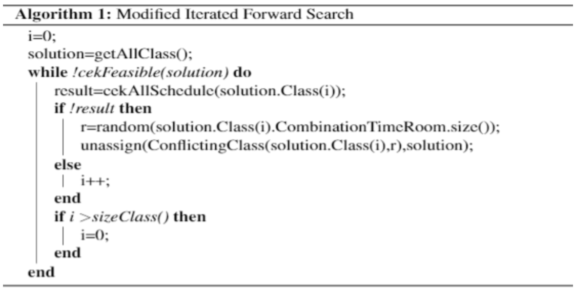
Figure 3. Third Elimination Flow

```
procedure SOLVE(initial)                // initial solution is the parameter
    iteration = 0;                      // iteration counter
    current = initial;                  // current solution
    best = initial;                     // best solution
    while canContinue(current, iteration) do
        iteration = iteration + 1;
        variable = selectVariable(current);
        value = selectValue(current, variable);
        UNASSIGN(current, CONFLICTING_VARIABLES(current, variable, value));
        ASSIGN(current, variable, value);
        if better(current, best) then best = current
    end while
    return best
end procedure
```

Figure 4. IFS algorithm pseudocode

Changes made in this study involve changing the selection of values, or in this case in the form of a schedule candidate from being randomly selected to trying each candidate schedule until it is found that there is no conflict with other variables or in this case a class. If it is found that there are no schedule candidates that can be used in a scheduled class, one of the schedule candidates will be selected randomly and the other classes that conflict with the schedule candidate will be returned to the unscheduled condition. The iteration will be repeated until it is found that all the class conditions have been scheduled and no one has violated the constraints. Figure 5 shows the changes made to the IFS algorithm.

To see if there is an impact of changes made to the IFS algorithm, this study runs and compares two algorithms, namely by using the IFS algorithm as shown in Figure 4 and the IFS algorithm which has been changed as in Figure 5.



```
Algorithm 1: Modified Iterated Forward Search
i=0;
solution=getAllClass();
while !cekFeasible(solution) do
    result=cekAllSchedule(solution.Class(i));
    if !result then
        r=random(solution.Class(i).CombinationTimeRoom.size());
        unassign(ConflictingClass(solution.Class(i),r),solution);
    else
        i++;
    end
    if i >sizeClass() then
        i=0;
    end
end
```

Figure 5. Pseudocode of IFS Algorithm Change Results

After obtaining the initial solution, optimization will be carried out using the Great Deluge algorithm with a hyper-heuristic approach. In the hyper-heuristic approach, there are two parts, namely move acceptance to choose whether or not a solution is accepted and low level heuristic (LLH) to change existing solutions.

In the move acceptance section, the Great Deluge algorithm is used. A new solution will be accepted if it produces a better result than the previous solution or if the solution is better than the level parameter value. The level parameter value is obtained from the initial solution value and will decrease continuously during the iteration. Figure 4 illustrates how this algorithm works.

In the LLH section, mutations are used to make changes to the solution. Mutations work by changing the schedule of one of the classes. Figure 7 illustrates the changes made by mutations. Initially there was a class 6 class that had been scheduled. Class 1 is scheduled for time slot 10, class 2 is scheduled for time slot 76, class 3 is scheduled for time slot 23 and so on. Mutations are applied to class 3 by scheduling at a different time slot, namely 27 time slots. The mutation only affects one randomly selected class. Meanwhile, other classes will not change.

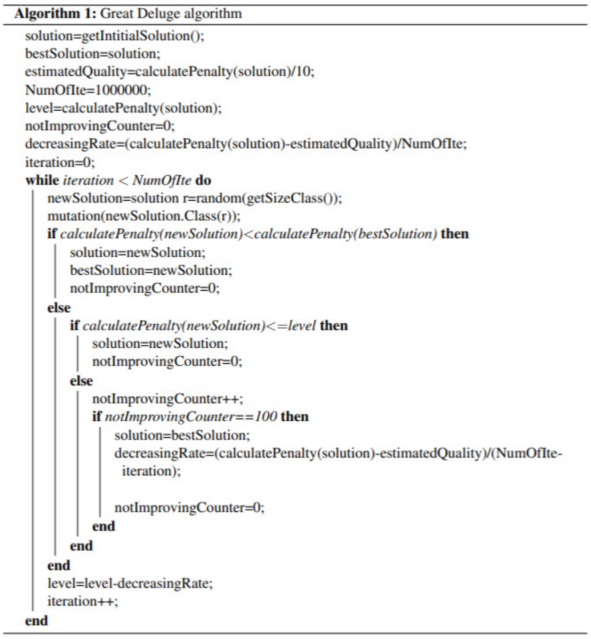
Furthermore, the algorithm design will be implemented through the Java programming language with a trial environment as shown in Table 3.

Table 3. Algorithm Implementation Environment

Device	Specification
Processor	AMD Ryzen 7 3700U (8 CPUs)
Ram	16GB
OS	Windows
Intellij	IDE 8.0.2

c. Final Solution Validation

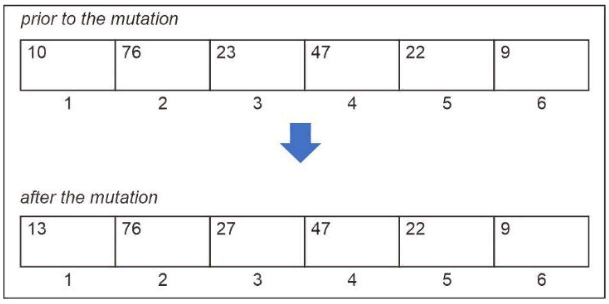
To ensure that the final result is a valid solution, the final result will be uploaded to https://www.itc2019.org/validator to check the solution. The validator will check whether any hard constraints are violated or not. In addition, the validator will calculate the penalty value so that the penalty value on the validator web can be compared with the penalty value that is owned in this test to ensure that the program of implementing the algorithm that has been designed produces the correct results. Solutions that have been deemed valid by the web validator will be stored on the website.



```
Algorithm 1: Great Deluge algorithm
solution=getIntitialSolution();
bestSolution=solution;
estimatedQuality=calculatePenalty(solution)/10;
NumOfIte=1000000;
level=calculatePenalty(solution);
notImprovingCounter=0;
decreasingRate=(calculatePenalty(solution)-estimatedQuality)/NumOfIte;
iteration=0;
while iteration < NumOfIte do
    newSolution=solution r=random(getSizeClass());
    mutation(newSolution.Class(r));
    if calculatePenalty(newSolution)<calculatePenalty(bestSolution) then
        solution=newSolution;
        bestSolution=newSolution;
        notImprovingCounter=0;
    else
        if calculatePenalty(newSolution)<=level then
            solution=newSolution;
            notImprovingCounter=0;
        else
            notImprovingCounter++;
            if notImprovingCounter==100 then
                solution=bestSolution;
                decreasingRate=(calculatePenalty(solution)-estimatedQuality)/(NumOfIte-
                 iteration);

                notImprovingCounter=0;
            end
        end
    end
    level=level-decreasingRate;
    iteration++;
end
```

Figure 6. Great Deluge Algorithm pseudocode



10	76	23	47	22	9
1	2	3	4	5	6

13	76	27	47	22	9
1	2	3	4	5	6

Figure 7. Examples of Mutation Operations in Scheduling

3. Results and Discussion

This section will explain the results obtained in this study.

a. Data Preprocessing

After the data preprocessing process was carried out, a reduction in schedule candidates was obtained in each dataset. Tables 3, 4 and 5 display the number of schedule candidates, the number of eliminated schedule candidates and the percentage of eliminations performed. Overall there is a reduction in schedule candidates by an average of 19.4% of the 30 datasets tested.

To see the impact of the data preprocessing process, a comparison was made between using the data preprocessing process and those that did not use the data preprocessing process. The result is that without the data preprocessing process, there are 8 datasets for which a solution cannot be found without violating existing hard constraints. The 8 datasets are Agh-fis-spr17, Nbi-spr18, Pu-d5-spr17, Pu-proj-fal19, Agh-fal17, Muni-fspsx-fal17, Muni-pdf-fal17 and Pu-d9-fal19.

The data preprocessing process has proven to be able to help find initial solutions without breaking existing hard constraints. This happens because if the schedule candidate is not eliminated in the data preprocessing process, the schedule candidate will interfere with the process of producing the initial solution. Schedule candidates who are unlikely to be used, can be selected in the process of generating the initial solution. This will have an impact on other classes and cause conflicts between classes.

b. Initial Solution and Final Solution

The initial solution was carried out after the data preprocessing stage by running the IFS algorithm which was adjusted to the 2019 ITC problem. The result was that this algorithm was able to produce an initial solution without breaking the hard constraints that existed in the entire dataset (30 datasets).

To see the impact of changes made to the IFS algorithm, two experiments were carried out using the original IFS algorithm and using the IFS algorithm that had been changed. The result is that using the original IFS algorithm without making any changes is only able to produce an initial solution on one dataset, namely the Mary-spr17 dataset. This happens because the original IFS algorithm selects schedule candidates randomly, so that the search for solutions is not well focused on finding solutions without breaking the existing hard constraints.

The next step is to optimize each dataset by running 1000,000 iterations. The result was that the largest penalty reduction was in the Lums-spr18 dataset, which was 78% and the smallest penalty reduction was in the bet-fal17 dataset, which was 7%. Overall, the average penalty reduction was 42%. Tables 6, 7 and 8 show the results of the comparison between the initial solution and the final solution after the penalty is applied.

In the Pu-proj-fal19, Bet-fal17, Agh-fal17, Bet-spr18, Muni-fspsx-fal17 and Muni-pdfx-fal17 dataset, optimization can only reduce penalties by a percentage of between 7-11%, especially in the late dataset. This happens because the late dataset is the most difficult dataset by having more classes, rooms and boundaries than the middle and early dataset groups. The Great Deluge algorithm cannot explore solutions on a dataset like this because when a solution changes, hard constraint violations often occur so that the new solution cannot be accepted.

Table 4. Preprocessing results on 10 datasets of early instances

Dataset	Number of Candidates Schedule	Number of Reduced Schedule Candidates	Percentage Reduction
Agh-fis-spr17	4373766	2715519	62%
Agh-ggis-spr17	353772	90790	26%
Bet-fal17	621508	101715	16%
Iku-fal17	3653791	102996	3%
Mary-spr17	169069	2177	1%
Muni-fi-spr16	48183	7371	15%
Muni-fsps-spr17	33810	2580	8%
Muni-pdf-spr16c	915725	43754	5%
Pu-llr-spr17	178424	47201	26%
Tg-fal17	77498	8116	10%

Table 5. Preprocessing results on 10 middle instance datasets

Dataset	Number of Candidates Schedule	Number of Reduced Schedule Candidates	Percentage Reduction
Agh-ggos-spr17	2331431	1155289	50%
Agh-h-spr17	2693914	1539477	57%
Lums-spr18	591475	199391	34%
Muni-fi-spr17	57843	8508	15%
Muni-fsps-spr17c	439903	84340	19%
Muni-pdf-spr16	915752	43754	5%
Nbi-spr18	144682	3000	2%
Pu-d5-spr17	86802	9991	12%
Pu-proj-fal19	1937292	259255	13%
Yach-fal17	93648	3857	4%

Preprocessing results on 10 late instance datasets

Dataset	Number of Candidates Schedule	Number of Reduced Schedule Candidates	Percentage Reduction
Agh-fal17	6012899	2574797	62%
Bet-spr18	649353	98402	15%
Iku-spr18	3588585	122101	16%
Lums-fal17	606929	220968	36%
Mary-fal18	187962	5295	3%
Muni-fi-fal17	46341	6374	14%
Muni-fspsx-fal17	579535	156832	8%
Muni-pdfx-fal17	5054749	1223449	24%
Pu-d9-fal19	677889	79274	12%
Tg-spr18	85330	11529	10%

c. Validate the final solution results

The final stage to ensure the results of the research are valid, validation of the final solution is carried out on the ITC 2019 website. The results are as in Figure 8, the 30 solutions produced have proven to be solutions that do not violate existing hard constraints by storing these solutions on the validator web. The penalty results for each solution are also stored on the website.

Table 6. Initial Solution and Final Solution on 10 dataset early instances

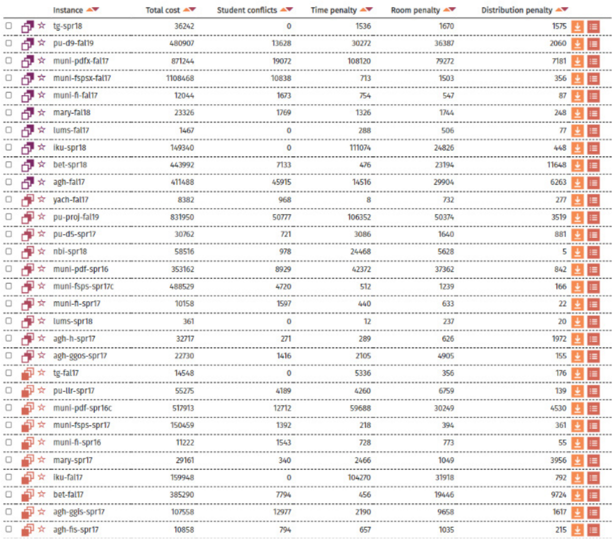
Dataset	Early Solution Penalty	Final Solution Penalty	Percentage Reduction
Agh-fis-spr17	38233	10858	72%
Agh-ggis-spr17	199432	107558	46%
Bet-fal17	414061	385290	7%
Iku-fal17	162211	112187	31%
Mary-spr17	77804	29161	63%
Muni-fi-spr16	24471	11222	54%
Muni-fsps-spr17	277511	150459	46%
Muni-pdf-spr16c	600126	517913	14%
Pu-llr-spr17	140771	55275	61%
Tg-fal17	26072	14548	44%

Initial Solution and Final Solution on 10 middle instance dataset

Dataset	Early Solution Penalty	Final Solution Penalty	Percentage Reduction
Agh-ggos-spr17	77059	22730	71%
Agh-h-spr17	55312	32717	41%
Lums-spr18	1638	361	78%
Muni-fi-spr17	23116	10158	56%
Muni-fsps-spr17c	657434	488529	26%
Muni-pdf-spr16	337279	212005	37%
Nbi-spr18	128733	58516	55%
Pu-d5-spr17	58746	30762	48%
Pu-proj-fal19	931627	831950	11%
Yach-fal17	29297	8382	71%

Initial Solution and Final Solution on 10 late instance datasets

Dataset	Early Solution Penalty	Final Solution Penalty	Percentage Reduction
Agh-fal17	552095	493834	11%
Bet-spr18	482804	443992	8%
Iku-spr18	198190	149340	25%
Lums-fal17	2695	1467	46%
Mary-fal18	59431	23326	61%
Muni-fi-fal17	30133	12044	60%
Muni-fspsx-fal17	1132578	1002804	11%
Muni-pdfx-fal17	952705	871244	9%
Pu-d9-fal19	627582	480907	23%
Tg-spr18	104276	36242	65%



	Instance	Total cost	Student conflicts	Time penalty	Room penalty	Distribution penalty
□ ☆	tg-spr18	36242	0	1536	1670	1575
□ ☆	pu-d9-fal19	480907	13628	30272	36387	2060
□ ☆	muni-pdfx-fal17	871244	19072	108120	79272	7181
□ ☆	muni-fspsx-fal17	1108468	10838	713	1503	356
□ ☆	muni-fi-fal17	12044	1673	754	547	87
□ ☆	mary-fal18	23326	1769	1326	1744	248
□ ☆	lums-fal17	1467	0	288	506	77
□ ☆	iku-spr18	149340	0	111074	24826	448
□ ☆	bet-spr18	443992	7133	476	23194	11648
□ ☆	agh-fal17	411488	45915	14516	29904	6263
□ ☆	yach-fal17	8382	968	8	732	277
□ ☆	pu-proj-fal19	831950	50777	106352	50374	3519
□ ☆	pu-d5-spr17	30762	721	3086	1640	881
□ ☆	nbi-spr18	58516	978	24468	5628	5
□ ☆	muni-pdf-spr16	353162	8929	42372	37362	842
□ ☆	muni-fsps-spr17c	488529	4720	512	1239	166
□ ☆	muni-fi-spr17	10158	1597	440	633	22
□ ☆	lums-spr18	361	0	12	237	20
□ ☆	agh-h-spr17	32717	271	289	626	1972
□ ☆	agh-ggos-spr17	22730	1416	2105	4905	155
□ ☆	tg-fal17	14548	0	5336	356	176
□ ☆	pu-llr-spr17	55275	4189	4260	6759	139
□ ☆	muni-pdf-spr16c	517913	12712	59688	30249	4530
□ ☆	muni-fsps-spr17	150459	1392	218	394	361
□ ☆	muni-fi-spr16	11222	1543	728	773	55
□ ☆	mary-spr17	29161	340	2466	1049	3956
□ ☆	iku-fal17	159948	0	104270	31918	792
□ ☆	bet-fal17	385290	7794	456	19446	9724
□ ☆	agh-ggis-spr17	107558	12977	2190	9658	1617
□ ☆	agh-fis-spr17	10858	794	657	1035	215

Figure 8. Validation Results

4. Conclusion

This research focuses on solving complex scheduling problems using the latest dataset from ITC 2019. The solution to this problem is carried out in two phases. The first phase is building an initial solution without breaking existing hard constraints. The second phase is optimizing the solution to make it better by reducing the number of penalties for violations of soft constraints.

The first phase is done by implementing the elimination of schedule candidates when preprocessing data and applying the IFS algorithm with some changes. As a result, all ITC 2019 datasets (30 datasets) were found to be viable initial solutions without breaking the hard constraints.

The second phase is optimization by applying the Great Deluge algorithm with a hyper heuristic approach using mutation LLH. The results obtained an average penalty reduction of 42%.

Reference

1. Lindahl, M., Mason, A. J., Stidsen, T. and Sørensen, “A strategic view of University timetabling,” *European Journal of Operational Research,* vol. 226, no. 1, pp. 35-45, 2018.
2. V. I. Skoullis, . I. X. Tassopoulos and G. N. Beligiannis, “Solving the high school timetabling problem using a hybrid cat swarm optimization based algorithm,” *Applied Soft Computing,* vol. 52, pp. 277-289, 2017.
3. M. Chen, X. Tang, T. Song, C. Wu, S. Liu and X. Peng, “A Tabu search algorithm with controlled randomization for constructing feasible university course timetables,” *Computers and Operations Research,* pp. 1-20, 2020.
4. J. S. Tan, S. L. Goh, G. Kendall and N. R. Sabar, “A survey of the state-of-the-art of optimisation

methodologies in school timetabling problems," *Expert Systems With Applications,* vol. 165, 2021.

5. T. Thepphakorn and P. Pongcharoen , "Performance improvement strategies on Cuckoo Search algorithms for solving the university course timetabling problem," *Expert Systems with Applications,* no. 161, 2020.
6. L. Saviniec and A. A. Constantino, "Effective local search algorithms for high school timetabling problems," *Applied Soft Computing,* vol. 60, pp. 363-373, 2017.
7. A. Rezaeipanah, S. S. Matoori and G. Ahmadi , "A hybrid algorithm for the university course timetabling problem using the improved parallel genetic algorithm and local search," *Applied Intelligence,* no. 51, p. 467–492, 2021.
8. T. Song, S. Liu, X. Tang, X. Peng and M. Chen, " An iterated local search algorithm for the University Course Timetabling Problem," *Applied Soft Computing,* vol. 68, pp. 597-608, 2018.
9. T. M¨uller, R. Bart´ak and H. Rudov´a, "Iterative Forward Search Algorithm: Combining Local Search with Maintaining Arc Consistency and a Conflict-Based Statistics," *Principles and Practice of Constraint Programming - CP,* vol. 3258, 2004.
10. Rudová, H., Müller, T. and Murray, K., "Complex university course timetabling," *Journal of Scheduling,* vol. 14, no. 2, p. 187–207, 2010.
11. A. Muklason, G. B. Syahrani and A. Marom, "Great Deluge Based Hyper-heuristics for Solving Real-world University Examination Timetabling Problem: New Data set and Approach," *The Fifth Information Systems International Conference 2019,* pp. 647-655, 2019.
12. T. M¨uller , ·. H. Rudov´a and Z. M¨ullerov´a, "University course timetabling and International Timetabling Competition 2019," *Proceedings of the 12th International Conference on the Practice and Theory of Automated Timetabling (PATAT-2018),* pp. 5-31, 2018.

khazanah informatika
Jurnal Ilmu Komputer dan Informatika

Implementation of the Fisher-Yates Shuffle Algorithm in Exam-Problem Randomization on M-Learning Applications

Chandra Kirana*, Benny Wijaya, Abdul Holil
Informatics Engineering Study Program
STMIK Atma Luhur
Pangkalpinang, Indonesia
*Chandra.kirana@atmaluhur.ac.id

Abstract-Many schools are currently using conventional approaches in learning material deliveries and examination methods. Conventional examination processes referred to here are the provision of question sheets in paper form. They have several drawbacks, such as students cheating and a waste of paper printing costs. To overcome these problems, we propose an online examination system. The online system leaves students to work on a different question set from other students. The feature is made possible by applying a randomization algorithm. There are several algorithms for scrambling questions, one of which is the Fisher-Yates Shuffle algorithm. This study aims to ease schools in the implementation of quality exams that may find out the level of student understanding of study materials and reduce the risk of cheating. The research product works on Android smartphones, which may be attractive to students and schools. The product allows schools to hold quality exams and reduce paper costs.

Keywords: Fisher-Yates shuffle, exam question, randomization, online exam

***Article info:** submitted August 4, 2020, revised December 1, 2020, accepted December 18, 2020*

1. Introduction

The use of information technology has now penetrated all fields, one of which is the field of education. In the world of education, it is necessary to improve the quality, speed, practicality, and convenience in various aspects. One of them is an exam that was previously conventional in nature, to be converted into an online exam. This is an effort to replace the old system with the overall goal of improving the existing system so that it can run more effectively and efficiently. School exams are activities carried out by educational units to measure the achievement of students' competencies, which in turn are used as a measure of achievement of the school[1]. In the world of education, the teacher is one of the spearheads in students' success. Teachers, thus, must be able to follow the development of information technology in order to be more optimal in teaching. The use of technology in making teaching materials and questions will improve the performance of teachers in providing services to students.

This study aims to create an Android-based application that can be used as a practice platform for doing exam questions. Each package of questions given must have a different order of questions from other question packages. This requires scrambling the questions. Many methods can be used to generate random numbers, one of which is the Fisher-Yates Shuffle algorithm which is an algorithm for generating random permutations from a finite set [2]. In addition, the Fisher-Yates Shuffle Algorithm also has a random function in forming a balanced random pattern. This Fisher-Yates Shuffle algorithm is used to randomize the questions in this study.

In general there are many randomization techniques that have been developed by different researchers which can be used in a variety of applications [3]. Among them is a research conducted by Abdi Suhazli who randomized the pieces of a picture in a puzzle game using the fisher yates shuffle method [4]. Research conducted by Widi Aulia Rohmah implements the fisher yates shuffle algorithm to randomize questions in a quiz game, where the questions that appear are randomized and the user does not easily guess the next question [5]. Another research was also conducted by Shafali Agarwal. In that research, the process of making a pixel permutation key was carried out using the knuth shuffle method and dynamic diffusion from random images [6]. Fyan Dimas Pratama's research in 2019 regarding the application of online daily tests using the Linear Congruential Generator (LCG) method concluded that this method can reduce question leakage [7]. A quite interesting research was also conducted by

Imam Haditama, Cepy Slamet and Deny Fauzy Rahman in 2016 regarding the Implementation of the Fisher-Yates and Fuzzy Tsukamoto Algorithm in an Android-Based Sundanese Tone Guess Quiz Game. The results of his research showed that the Fisher-Yates algorithm is able to determine the solution of non-multiple randomization and multiple object randomization [8]

In this study, the authors used the fisher yates shuffle algorithm for the randomization process because the algorithm uses a shuffling technique used to change the order of an element randomly where the randomization process is unbiased and random [9]. Though the algorithm has a dynamic shuffling nature, the modern form of the algorithm is more efficient. The implementation enhances the time complexity to $O(n)$, from $O(n^2)$. Here, the rolled number indicates the position of the element removed from the current list and inserted into the shuffled list and number at the end of the current list is placed at the same position that yields further shuffling at each iteration. This makes the system more dynamic. When the list contains one element, it is removed and inserted into shuffled list [10].

2. Method

In this study, the authors created a framework to facilitate the development. This framework can be seen in Figure 1.

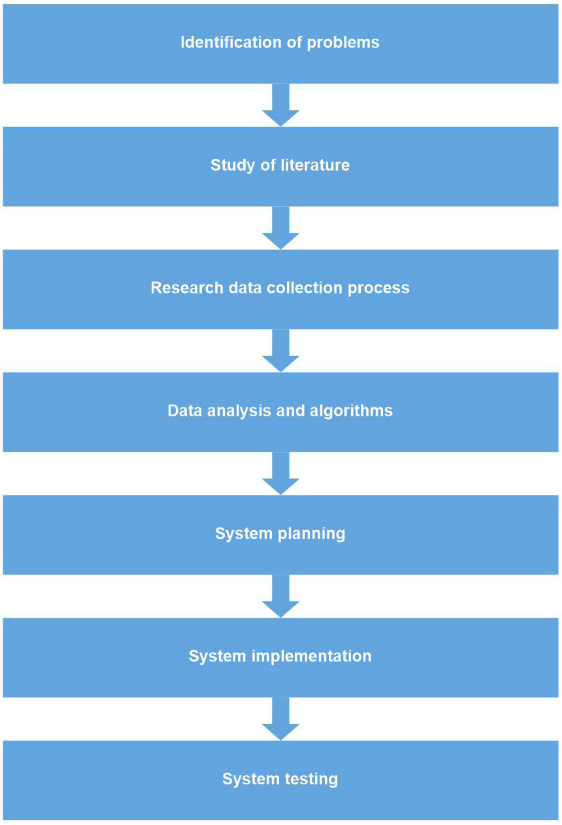

Identification of problems
Study of literature
Research data collection process
Data analysis and algorithms
System planning
System implementation
System testing

Figure 1. Research Framework

The description is as follows

1. Problem Identification
   The process carried out at this stage is to formulate the problems that occur today and which are the objects in the research. The formulated problem is related to the process of randomizing questions in the implementation of the exam and testing the Fisher-Yates Shuffle algorithm in randomizing questions..
2. Literature Review
   This stage is then the authors perform data analysis on the problems that occur. One of the analysis processes carried out by the author is by studying the literature related to the problem tracking algorithm. The data source is obtained by the author through journals and books related to question randomization and the Fisher-yates shuffle algorithm.
3. Data collection
   At this stage the writer carried out the data collection process where the writer took the data sample by means of interviews, observations, and observations.
4. Data analysis
   At this stage, the authors carry out the process of analyzing the data needed in this study, such as data for the randomization process and data related to the Fisher-yates shuffle algorithm..
5. System planning
   At this stage the authors carry out the system design process to be built including the desired form of system design and planning in making the system. The purpose of this design is to facilitate the next process, namely at the stage of making the desired system.
6. System Implementation
   After the system design process is carried out, the next step is to build the desired system.
7. System Testing
   This stage is where the writer ensures whether the process previously carried out is running as expected or not.

3. Result

a. Data analysis

The analysis in this study includes how the fisher-yates shuffle algorithm works in the process of randomizing the questions that are applied in the online exam system that has been built. An overview of the system can be seen in Figure 2.

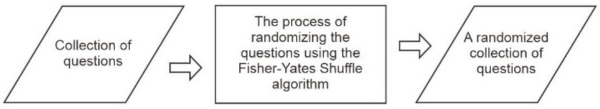

Collection of questions
The process of randomizing the questions using the Fisher-Yates Shuffle algorithm
A randomized collection of questions

Figure 2. System Overview

b. Analysis of the Fisher-Yates Shuffle Algorithm

This development is one way that can be done to minimize cheating in doing practice questions. The process of implementing the algorithm can be seen in Figure 3.

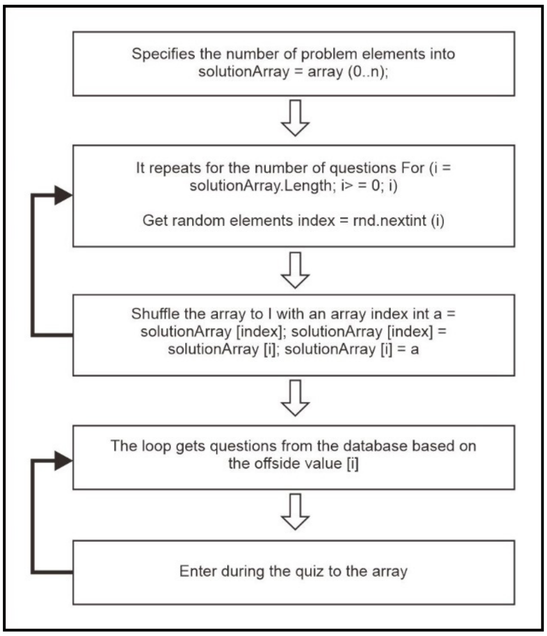

Specifies the number of problem elements into solutionArray = array (0..n);
It repeats for the number of questions For (i = solutionArray.Length; i> = 0; i)
Get random elements index = rnd.nextint (i)
Shuffle the array to I with an array index int a = solutionArray [index]; solutionArray [index] = solutionArray [i]; solutionArray [i] = a
The loop gets questions from the database based on the offside value [i]
Enter during the quiz to the array

Figure 3. Fisher-Yates Shuffle Algorithm Process Flow

The steps used to generate a random permutation for problems 1 to *N* can be seen as follows [3]:

1. Write down the questions from number 1 to number *N*.
2. Choose a random question *K* between 1 and the number of questions that have not been crossed out.
3. Count from the bottom position. Cross out the *K* questions that have not been crossed out and write the questions in another place.
4. Repeat step 2 and step 3 until all the questions have been crossed out.
5. The order of the questions written in step 3 is the random permutation of the initial problem.

Table 1. Randomized questions using the Fisher-Yates Algorithm

Range	Roll	Scratch	Result
		1,2,3,4,5,6,7,8,9,10,11,12,13,14,15	
1-15	6	1,2,3,4,5,7,8,9,10,11,12,13,14,15	6
1-14	10	1,2,3,4,5,7,8,9,10,12,13,14,15	11,6
1-13	3	1,24,5,7,8,9,10,12,13,14,15	3,11,6
1-12	1	2,4,5,7,8,9,10,12,13,14,15	1,3,11,6
1-11	6	2,4,5,7,8,10,12,13,14,15	9,1,3,11,6
1-10	4	2,4,5,8,10,12,13,14,15	7,9,1,3,11,6
1-9	7	2,4,5,8,10,12,14,15	13,7,9,1,3,11,6
1-8	7	2,4,5,8,10,12,15	14,13,7,9,1,3,11,6
1-7	2	2,5,8,10,12,15	4,14,13,7,9,1,3,11,6
1-6	4	2,5,8,12,15	10,4,14,13,7,9,1,3,11,6
1-5	5	2,5,8,12	15,10,4,14,13,7,9,1,3,11,6
1-4	4	2,5,8	12,15,10,4,14,13,7,9,1,3,11,6
1-3	3	2,5	8,12,15,10,4,14,13,7,9,1,3,11,6
1-2	2	2	5,8,12,15,10,4,14,13,7,9,1,3,11,6
			2,5,8,12,15,10,4,14,13,7,9,1,3,11,6

The next step is to enter the question attribute into scratch (a list of questions that have not been selected) then create a range (the number of questions that have not been selected). Next do the randomization process, and then see the roll (for a question that is selected from all the existing questions). After that, the results of the questions that have been selected are entered into the results (the results of all the questions that have been randomized) [2]. The process of implementing the randomization algorithm for fifteen questions is exemplified in Table 1.

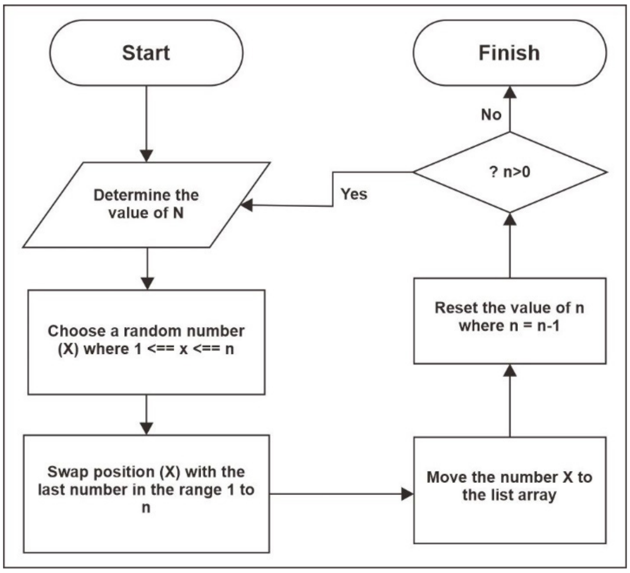

Start
Finish
No
? n>0
Yes
Determine the value of N
Choose a random number (X) where 1 <== x <== n
Reset the value of n where n = n-1
Swap position (X) with the last number in the range 1 to n
Move the number X to the list array

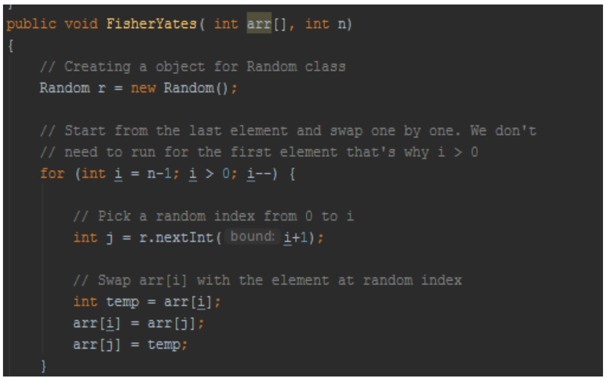
Figure 4. The fhiser yates shuffle randomization method [3]

From Figure 4, it can be seen that the randomization is complete when the entire array has been scrambled. The fisher-yates shuffle randomization method produces a randomized array sequence.

c. System Testing

This research resulted in the implementation of the fisher yates shuffle algorithm which is used for the randomization process in mobile-based M-Learning applications. Algorithm testing is carried out on practice exam questions, thus each student will get a different order of questions from other students.

Algorithm implementation into the M-Learning application can be seen in Figures 5 and 6.



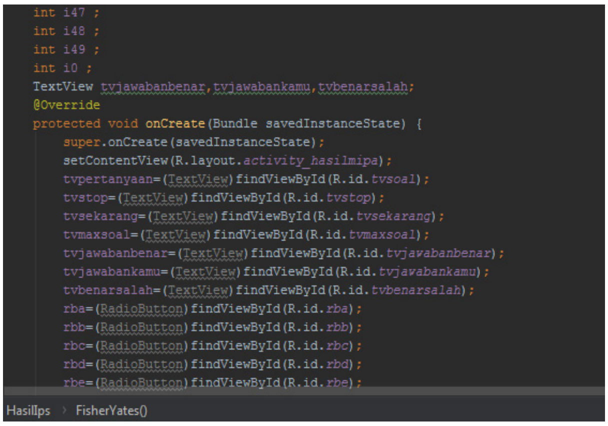
```
public void FisherYates( int arr[], int n)
{
    // Creating a object for Random class
    Random r = new Random();

    // Start from the last element and swap one by one. We don't
    // need to run for the first element that's why i > 0
    for (int i = n-1; i > 0; i--) {

        // Pick a random index from 0 to i
        int j = r.nextInt( bound: i+1);

        // Swap arr[i] with the element at random index
        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
    }
```

Figure 5. The process of selecting Random Problem



```
    int i47 ;
    int i48 ;
    int i49 ;
    int i0 ;
    TextView tvjawabanbenar,tvjawabankamu,tvbenarsalah;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_hasilmipa);
        tvpertanyaan=(TextView)findViewById(R.id.tvsoal);
        tvstop=(TextView)findViewById(R.id.tvstop);
        tvsekarang=(TextView)findViewById(R.id.tvsekarang);
        tvmaxsoal=(TextView)findViewById(R.id.tvmaxsoal);
        tvjawabanbenar=(TextView)findViewById(R.id.tvjawabanbenar);
        tvjawabankamu=(TextView)findViewById(R.id.tvjawabankamu);
        tvbenarsalah=(TextView)findViewById(R.id.tvbenarsalah);
        rba=(RadioButton)findViewById(R.id.rba);
        rbb=(RadioButton)findViewById(R.id.rbb);
        rbc=(RadioButton)findViewById(R.id.rbc);
        rbd=(RadioButton)findViewById(R.id.rbd);
        rbe=(RadioButton)findViewById(R.id.rbe);
```

Figure 6. The process of displaying the results of scrambling questions

The results of the randomization test with the Fisher Yates shuffle algorithm can also be seen in Table 2.

Table 3. Result of randomization test

Testing to-	The result of the appearance of the question number
1	37,16,39,11,7,47,10,45,31,9,13,36,35,26,4,38,28,46,27,14,19,12,6,41,48,42,15,29,30,17,1,22,40,5,32,8,20,43,24,50,33,18,2,21,23,3,44,25,34,49
2	21,30,34,26,36,24,39,8,17,29,25,3,14,42,49,47,13,32,50,1,27,35,28,16,48,10,23,15,2,7,41,33,4,46,5,38,9,18,19,45,43,28,44,12,6,11,22,40,37,20
3	18,16,20,6,38,36,39,40,7,29,32,30,31,44,13,8,14,37,27,24,2,42.9,12,46,19,23,26,11,35,17,48,33,28,22,43,15,49,41,21,10,3,47,50,44,1,25,5,34,45
4	5,21,27,28,42,24,47,1,19,37,25,22,43,29,18,39,35,6,13,49,3,16,11,40,14,48,41,32,46,15,12,30,34,17,38,31,20,36,45,2,7,50,10,44,4,9,8,23,33,26
5	33,3,48,49,14,30,6,40,31,39,46,36,16,18,10,34,28,29,25,8,19,22,23,38,4,9,32,26,43,41,20,2,24,12,47,5,37,11,21,50,17,1,7,35,45,15,27,13,42,44

In Figure 7, it can be seen that the fisher yates shuffle algorithm was successfully applied to the M-Learning application by using several cellphones simultaneously with different questions.

Tests in this study used two tests, namely alpha and beta testing. Alpha testing is also known as blackbox testing. This test is carried out using software specifications without referring to procedures from the internal system being made. The purpose of this test is to test whether the components in the system are in accordance with the desired design. This blackbox test only looks at the basics of the system and ensures that the input made can be accepted by the system properly and produces output as expected. The testing process can be seen in Table 3.

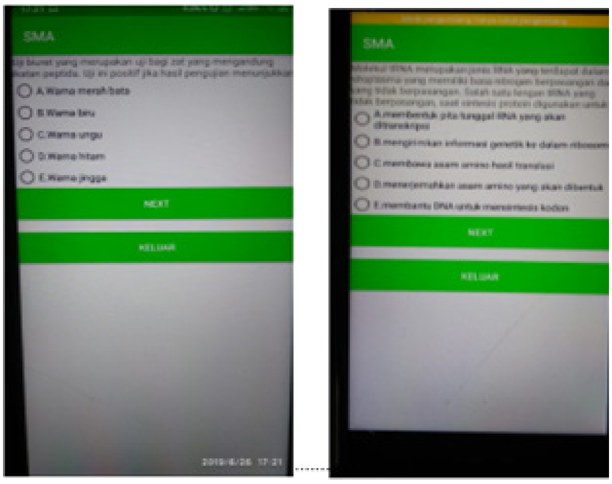


Figure 7. Algorithm testing on the M-Learning application

Table 3. Black Box Test Results

Activity	Output	Result
Login	The system displays the main page	According to the design
List	The system displays a list form and stores data into the database	According to the design
Select Material	The system displays the Material page	According to the design
Select Problem	The system displays the Questions page	According to the design

Based on Table 3, it can be seen that all system functions that are built can run well using alpha testing (blackbox). If in system use an error occurs it is caused by a system user who does not provide the correct input. In addition to Alpha testing, this study also uses Beta testing, namely testing using a questionnaire. The test sample is as many as 20 individuals, consisting of 15 students and 5 teachers. The list of questionnaire results from this test can be seen in Table 4.

Table 4. Beta Testing Results with a Questionnaire

Questionnaire	Answer (%)			
	1	2	3	4
Do you agree, if the system that is made has a good user interface?	0%	0%	0,45%	0,55%
Do you agree, if the system created can make it easier for the school in the learning process, especially the exam process?	0%	0%	0,3%	0,7%
Do you agree, if this system makes it easy to use?	0%	0%	0,25%	0,75%

Questionnaire	Answer (%)			
	1	2	3	4
Does Ands agree, if the system created can randomize the questions, so that the questions differ from one another?	0%	0%	0,1%	0,9%

Note: 1: Strongly disagree; 2: Disagree; 3: Agree; 4: Totally agree

The results of this Beta test can be concluded as follows: (1) The system created has a good user interface and is easy to use; (2) Users can follow and use the system in accordance with the instructions provided; (3) This system is able to provide different questions from one another by applying the Fisher-Yates Shuffle algorithm as a process of randomizing the questions..

4. Conclusion

This M-Learning system has been built with the Android operating system. While the development of the system uses a prototype model and development tools using UML. For testing, this study uses Beta and Alpha testing. The results of Alpha testing using the Black box show that all functions in the M-Learning system can work properly in accordance with the system design, especially for the process of randomizing questions using the Fisher-Yates Shuffle algorithm. Beta test results can be concluded as follows: (1) The system created has a good user interface and is easy to use; (2) Users can follow and use the system in accordance with the instructions provided; (3) This system is able to provide different questions from one another by applying the Fisher-Yates Shuffle algorithm as a process of randomizing the questions. Presentation of the results of this system using mobile technology, which is based on Android, so that the system used can provide convenience in its use and where and whenever students can carry out the learning process without having to go to school.

Reference

[1] Gunawan And D. A. Prabowo, “Sistem Ujian Online Seleksi Penerimaan Mahasiswa Baru Dengan Pengacakan Soal Menggunakan Linear Congruent Method (Studi Kasus Di Universitas Muhammadiyah Bengkulu),” *Inform. Upgris*, Vol. 3, No. 2, Pp. 143–151, 2017.

[2] M. A. Hasan, Supriadi, and Zamzami, “Implementasi Algoritma Fisher-Yates Untuk Mengacak Soal Ujian Online Penerimaan Mahasiswa Baru (Studi Kasus : Universitas Lancang Kuning Riau),” *J. Nas. Teknol. dan Sist. Inf.*, vol. 03, no. 02, pp. 291–298, 2017.

[3] S. A., S. N., and M. N., “Generating Random Data using 3 Nonlinear Functions,” *Int. J. Comput. Appl.*, vol. 152, no. 4, pp. 6–10, 2016, doi: 10.5120/ijca2016911776.

[4] A. Suhazli, A. Atthariq, and A. Anwar, “Game Puzzle ‘Numbers in English’Berbasis Android Dengan Metode Fisher Yates Shuffle Sebagai Pengacak Potongan Gambar,” *J. Infomedia*, vol. 2, no. 1, pp. 1–6, 2017, doi: 10.30811/.v2i1.476.

[5] W. A. Rohmah, A. Asriyanik, and W. Apriyandari, “Implementation of the Algorithm Fisher Yates Shuffle on Game Quiz Environment,” *J. Informatics Telecommun. Eng.*, vol. 4, no. 1, pp. 161–172, 2020, doi: 10.31289/jite.v4i1.3863.

[6] S. Agarwal, “A fractal based image cipher using Knuth shuffle method and dynamic diffusion,” *Int. J. Comput. Networks Commun.*, vol. 11, no. 4, pp. 81–100, 2019, doi: 10.5121/ijcnc.2019.11405.

[7] F. D. Pratama, “APLIKASI ULANGAN HARIAN ONLINE Menggunakan Metode Linear Congruential Generator Berbasis Website,” Universitas Teknologi Yogyakarta, 2019.

[8] I. Haditama, C. Slamet, and D. F. Rahman, “IMPLEMENTASI Algoritma Fisher-Yates Dan Fuzzy Tsukamoto Dalam Game Kuis Tebak Nada Sunda Berbasis Android,” Vol. I, No. 1, Pp. 51–58, 2016.

[9] M. Tayel, G. Dawood, and H. Shawky, “Block cipher S-box modification based on fisher-yates shuffle and ikeda map,” *Int. Conf. Commun. Technol. Proceedings, ICCT*, vol. 2019-October, pp. 59–64, 2019, doi: 10.1109/ICCT.2018.8600161.

[10] T. K. Hazra, R. Ghosh, S. Kumar, S. Dutta, and A. K. Chakraborty, “File encryption using Fisher-Yates Shuffle,” *2015 Int. Conf. Work. Comput. Commun. IEMCON 2015*, no. March 2016, 2015, doi: 10.1109/IEMCON.2015.7344521.

khazanah informatika
Jurnal Ilmu Komputer dan Informatika

Word Cloud of UKSW Lecturer Research Competence Based on Google Scholar Source

SuryasatriyaTrihandaru, Hanna Arini Parhusip*, Bambang Susanto, Carolina Febe Ronicha Putri
Postgraduate Study in Data Science, Faculty of Science and Mathematics
Universitas Kristen Satya Wacana
Salatiga
*hanna.parhusip@uksw.edu

Abstract-There is a need in the Universitas Kristen Satya Wacana (UKSW) to identify the research competence of their faculties at a study program and University level. To accomplish this requirement, we need to automate the analysis of research output and publications quickly. Research articles are scattered in many publisher systems and journals which may be reputable, unreputable, accredited, and unaccredited. We devise a computer code to quickly and efficiently retrieve publication titles recorded in Google Scholar using a machine learning algorithm. The result display is in the form of a word cloud so that dominant and frequent words will be prominent in the visualization. In determining scientific terms to display, we used a modified version of the word cloud Python module and unmodified Term Frequency - Inverse Document Frequency (TF-IDF) library. The algorithm was tested on publication titles of our study program in UKSW and confirmed directly. The system features the ability to produce a word cloud visualization for an individual faculty, for faculties in a study program, or in the University as a whole. We have not differentiated publication sources, whether they are reputable or unreputable, which might affect the accuracy of competence identification.

Keyword*:* machine learning, word cloud, corpus, research competence

Article info: *submitted December 31, 2020, revised April 4, 2021, accepted April 19, 2021*

1. Introductions

Efforts to collect data on research results at a university often experience difficulties because the data is not documented in an integrated manner. Universitas Kristen Satya Wacana (UKSW) observed this situation. In addition, to trace the research competence of lecturers, university management cannot automatically search for documents to be able to make global conclusions where the documents are scattered in several groups known as reputable (international) and national journals, accredited and non-accredited journals. In addition, data changes over time, too, cannot be followed quickly. Therefore, a technique is needed to be able to automate reading and classifying lecturer research documents.

One of the used techniques to automate digital documents in research is to classify text [1]. In the literature, classification techniques are compared to study the accuracy of several classification techniques in data classification in the form of text but it is still not easy to read visually. For this reason, in this study, the Word cloud was used to be able to provide classification results more easily where this method has also been done by other authors in analyzing text using Latent Dirichlet allocation [2].

Word cloud is a machine learning algorithm. Machine Learning (ML) has been used in various applications in big data processing such as in the manufacture of artificial intelligence, processing COVID-19 data on the death rate in South Korea [3], diabetes data processing [4], and various other applications. Machine Learning (ML) is translated as machine learning in this article as part of data collection, for example, the study of large amounts of text analysis (big data) [5], pattern recognition, and computational theory in artificial intelligence for the initial process of analyzing skin images and the selection of Melanoma features by staining extraction [6] which builds on the theory, method and application of domains related to big data [7]. Likewise, ML is also often related to data mining where data mining explores text data analysis [8]. As a preliminary research, the Word Cloud algorithm was studied to collect spam and non-spam emails [9]. The machine will remember how based on previous knowledge that the email was said to be spam by the user, the next incoming email could be classified as spam and not spam. Such a working approach is called 'learning by reminder'.

This has a deficiency in the learning aspect, namely the ability to label invisible e-mail messages. A learning is successful if it can make progress individually in making wider links. To reach the link in the task of filtering emails on spam, students can search for previously viewed emails and extract words in messages that are indicated as spam. When a new email arrives, the engine can test whether the words in the email are spam and suspect the label [9]. Such a system can correctly predict labeling in invisible emails.

Based on this knowledge, the Universitas Kristen Satya Wacana (UKSW) lecturer research data was then classified, which is expected to be done automatically. The urgency of this research is shown by the need for universities to identify the competence of lecturers in each study program, each faculty and demonstrate the university's excellence through regularly documented research data that can be reported easily where in November 2020 the leadership needs the results of this research immediately. By using the classification techniques that have been studied for spam and non-spam e-mails, this article explains the results of research on UKSW lecturers' research data based on google scholar data, in which more than 2 classifications are formed based on the Word cloud. Lecturers' research data are classified into study program groups and faculties where all article title information in reputable and unreputable journals as well as accredited and unaccredited is not separated.

2. Method

a. Word cloud creation stage

In the literature, there are several Word cloud generators. However, the existing algorithms need to be adapted to the needs of this study. In principle, the font size of a word in the Word cloud is determined by the frequency with which it occurs. For lower frequencies, the font size can be used immediately. Say the initial size is s_0. For larger frequency values, the letters are scaled, normalized linear. Let the value of t_i be the *i-th* count, t_{max} is the maximum count, while t_{min} is the minimum count, then the font size is scaled in the formulation [10]which gains increasing attention and more application opportunities as the big data time approaches. Currently, there has been some online word cloud generators available for users with simple requests, such as repeating the exact phrase, or collecting the text data from a web page. Moreover, most current word cloud generators cannot support characters other than English, which are limited in English-speaking users. There are also packages for programming languages (such as Python and R :

$$s_i = \begin{cases} \left[\dfrac{f_{max}(t_i - t_{\min})}{(t_{max} - t_{min})} \right], t_i > t_{min} \\ 1 \quad , \quad \text{otherwise.} \end{cases}$$

However, this formula has been formulated in the Word cloud generators in Python so that the author doesn't need to do the formulation. The used algorithm is the NLP (Natural Language Process) algorithm, which is a branch of artificial intelligence that is focused on enabling computers to understand and interpret human language. The NLP process is demonstrated at the following stages.

1) Input data

At this stage, the data are inputted by adjusting the referred system,i.e. data from all UKSW lecturers, whether they have Google Scholar ID or not. The titles data from all articles in the google scholar of each lecturer until July 2020 are documented with the help of the Python program so that all titles are collected automatically.

2) Preprocessing data

The first collected data were titles of research articles that were detected on google scholar from UKSW lecturers who have google-scholar IDs. In this step, we transform these data into a recognizable format for the NLP model. Data are usually incomplete, inconsistent, and/or lacking something or trend and also contain errors. Among them, the data contain several incorrectly written words.

a) Tokenization Steps

The tokenization step is the process of cutting words/text into meaningful words, phrases, or elements called tokens. The steps in tokenization in English are shown in the literature [11] where in this study the tokenization for word lists in English and Indonesian. The token list is then used as a further process. The nltk library has tokenized words to make lists of sentences to be split into words or sentences.

b) Lemmatization Step

The Lemmatization step or Word Stemming has the same objective as the above process, namely reducing the inflected forms of each word to the basic form or root words whose relevance is appropriate to the user or reader [12]. For example: the word eating becomes eat. Lemmatization is close to stemming: the difference is that the stemmer operates a word without any knowledge of the context and therefore cannot distinguish words that have different meanings (e.g. bread eaten and eaten bread are clearly different, but here both are not distinguished). Stemmers are usually easier to implement and faster and the drop inaccuracy can be insignificant for some applications. However, because the data are in the titles of articles, this lemmatization was not carried out in this study because the titles were considered singular or unique.

The following are the complete steps that make up the preprocessing stages:

(a) Removes blank rows from the data if any

(b) Changes all letters in lowercase

(c) Token words
(d) Remove the stop word

As mentioned above, lemmatization was not performed in this case.

3) Prepare training data and test data
As in the Machine learning step, we need to separate the data into training data and test data. Training data are taken from some data that have been prepared where direct communication has obtained information on the characteristics of each lecturer or each study program from the existing data. The training data are carried out by the Word cloud in accordance with the information provided.

b. Example of classification testing with the Word cloud for spam and non-spam e-mails

As the first step in research, it is necessary to study word cloud on spam and non-spam email data obtained from the internet where the problem of processing text data for email is also an issue that researchers often study [9]. For that reason, you need the Word cloud generator so you need to install Word cloud. If on anaconda, we can install it on the anaconda menu, namely: pip install wordcloud. Furthermore, 4 DataFrames are created, namely:

a. DataFrame token_vek, contains tokens for each email that has been cleared, in a table of the number of tokens per email. This becomes the feature of the given data.
b. Class_vek dataFrame, contains spam or non-spam classification vectors.
c. Email_vek DataFrame, contains email tokens that have been strung together, for each email.
d. DataFrame vokab_vek, contains the vocabularies used to construct the DataFrame token_vek table.

The word cloud results provide layout, size, and color settings where the word with the greatest frequency will appear the largest and the color is more dominant so that it is easily recognized.

Figure 1 is a Word cloud on the classification of emails in spam. Whereas Figure 2 is a Word cloud in the classification of email which is classified as non-spam. Meanwhile, Figure 3 combines the results of spam and non-spam emails.

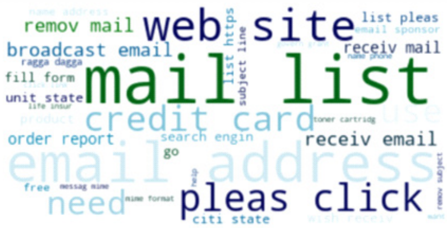


Figure 1. Word cloud results for spam classified emails with data retrieved from the internet https://spamassassin.apache.org/.

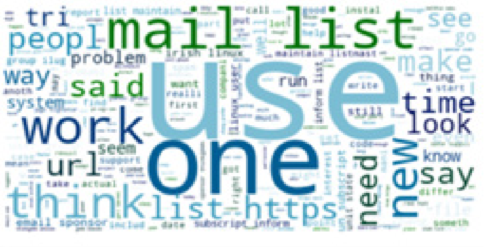


Figure 2. Word cloud results for email classified as non-spam with data taken from the internet https://spamassassin.apache.org/

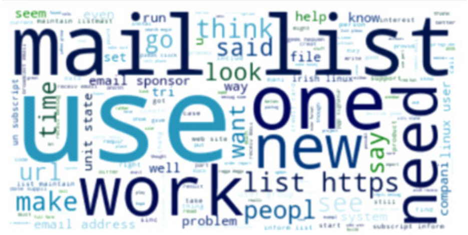


Figure 3. Word cloud results for email classified as spam and non-spam with data retrieved from the internet https://spamassassin.apache.org/.

c. Corpus used

Corpus studies the use of real-life language based on text, describing quantitative and qualitative text analysis techniques. In processing Indonesian text data, programs in the R language can directly accommodate changes from English to Indonesian for the case of text data [13]. In the Word Cloud algorithm that is already in Python, there is already Corpus which is commonly used in text analysis. At first, using the corpus which is shown in the scholarly function. However, this Corpus needs to be modified in this study by defining the Stop-word in this study. This is because the used data are the titles of UKSW lecturer articles that are spread on Google Scholar, from reputable, unreputable journals, accredited and unaccredited journals. So Corpus was built from these research titles indicated on Google Scholar. Therefore, Stop-word is also made in-house where the program is tested several times to improve Stop-word. The words that are considered as Stop-words are shown below.

Case 1.
Words that often appear in various lecturers, study programs, or faculties. The words ‘making’, ‘study’, ‘planning’, ‘learning’, ‘method’ are words that are not unique or special so they need to be discarded.

Case 2.
The word that is mistyped is also a word that needs to be listed in the stop word. For example “studen” is found in the data it needs to be a list of stop words.

Case 3.
Characters such as $,#,@, :, ; , are non-unique words that need to be removed.

It should be noted that the data contain journal titles both in Indonesian and in English. To identify the uniqueness of each lecturer or in the study program group or faculty, words that often appear are certainly not unique. Therefore, the corpus is made separately according to the needs of this research where the data of lecturers must have ID-scholars. Lecturers who do not have ID-scholar cannot contribute to this research.

d. Word Vectorization with TF-IDF

This step is a general process that converts a collection of text documents into feature vectors. There are many methods for this, but the popular one is called TF-IDF ("Term Frequency — Inverse Document" Frequency) which assigns a score to each word. TF-IDF is used directly from the word cloud generator in Python where in general TF-IDF is to summarize how often a word appears in a document (TF) and provide scaling (in descending order) to words that appear in a document (IDF) [14] . So the TF-IDF formulation is not defined by the researcher but directly uses the word cloud generator in Python which contains the TF-IDF formulation. One of the uses of TF-IDF in document classification is shown by a researcher in classifying documents in the types of politics, agriculture, economy, performance, science, and technology [15]. After the corpus is formed as described above, the TF-IDF builds a collection of words that have been learned from the corpus data and which designates a single integer to these words.

3. Result and Discussion

As mentioned in the preprocessing process, the use data are data on titles of articles from UKSW lecturers' research results from Google Scholar. For that, the researchers first recorded all the data of UKSW lecturers, for those with Google Scholar IDs and those without IDs. This process leads to obtaining 390 lecturers with ID-google scholar. With the Python program following the method above, all the titles of articles that have been documented by Google Scholar can be detected by the program. To test the program for this data, The Word cloud was conducted for the Master of Data Science lecturers where the author of this study was also part of the data.

a. Case data Word cloud lecturer Master of Data Science

As a studied initial case, the research data are from the Data Science Master Program lecturers at the UKSW Science and Mathematics Faculty (FSM). By using the method described in Chapter 2, the results of the Word cloud are obtained according to Figure 4-8. As in the explanation of the method, the TF-IDF is not explicitly defined by the author but has been defined directly in the word cloud generator in python. In this initial study, the results of the word cloud were obtained and then communicated directly to the author on the appearance of the word cloud obtained so that TF-IDF was not carried out further studies considering time constraints and the need for immediate word cloud results for each lecturer as well as each study program and faculty must be reported immediately in November 2020. Therefore, accuracy testing is carried out by directly confirming several lecturers regarding the data in the word cloud. In this case, preliminary research was conducted for 5 Data Science Masters lecturers at the UKSW FSM. The obtained information shows that it is appropriate to the research results. Therefore the research is continued by using data for all UKSW lecturers.

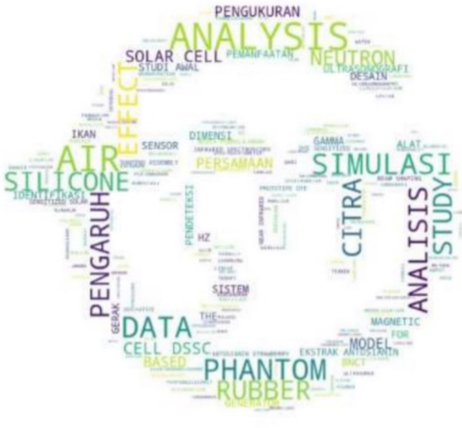


Figure 4. Word cloud for research data, Dr. Suryasatriya Trihandaru, S.Si., MSc.nat based on data up to July 2020.

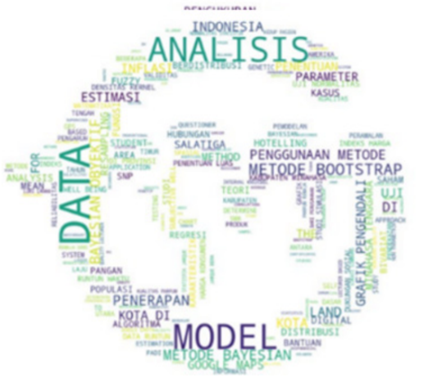


Figure 5. Word cloud for research data, Dr. Adi Setiawan, MS-based on data up to July 2020.

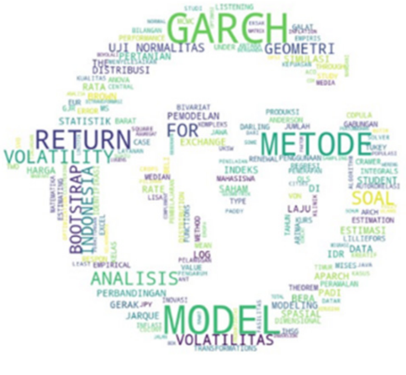


Figure 6. Word cloud for research data, Didit Budi Nugroho, MSi,DSc based on data up to July 2020.

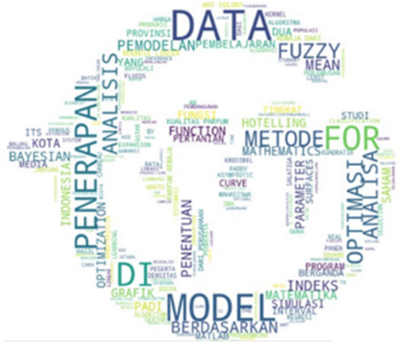


Figure 7. Word cloud for research data, Dr. Hanna Arini Parhusip based on data up to July 2020.

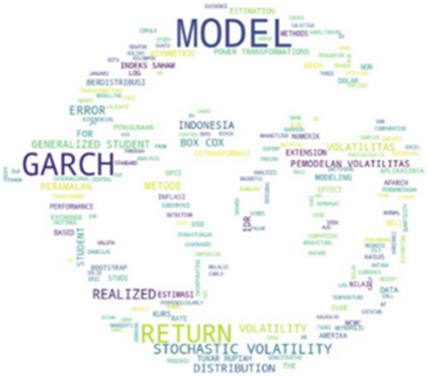


Figure 8. Word cloud for research data, Dr. Bambang Susanto, MS based on data up to July 2020.

b. How to do the analysis?

So far, the Word cloud has not been studied for the accuracy of the model obtained. However, at a glance, the words that stand out from the 5 samples are the word “Model” and the word “Data”. Another dominant word is ‘Analysis’. Broadly speaking, it can be concluded that the research shown by the 5 samples in Figure 4-8 above is about data modeling and analysis. One word that stands out is Garch. From the journal status that appears, it is known that articles with the word “Garch” are included in Scopus, thus gaining dominance in the Word cloud. The next case study will search UKSW’s leading research using the Word cloud for all UKSW lecturers through this research.

c. UKSW Word cloud case data research results

The search for UKSW leading research has been carried out using the Word cloud data for all UKSW lecturers in each of the UKSW faculties. By using machine learning according to the method in Chapter 2, the results of the Word cloud in several faculties are obtained as shown in Figure 9-15.

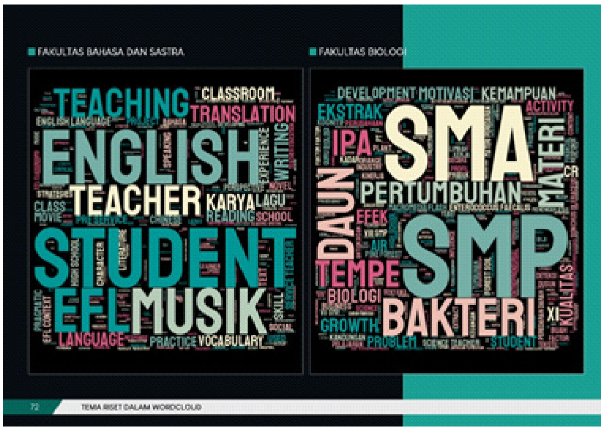


Figure 9. Word cloud for research data from the Faculty of Languages and Letters (left) and the Faculty of Biology (right)

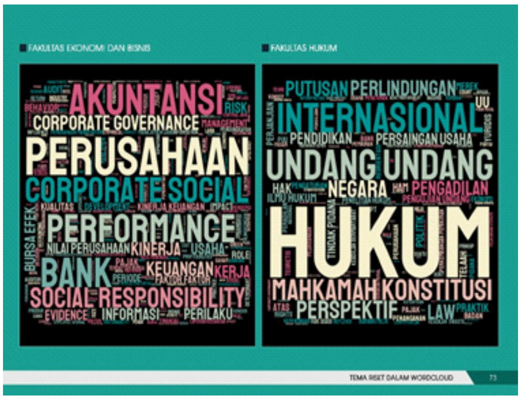


Figure 10. Word cloud for research data from the Faculty of Economics and Business (left) and the Faculty of Law (right) with data up to July 2020.

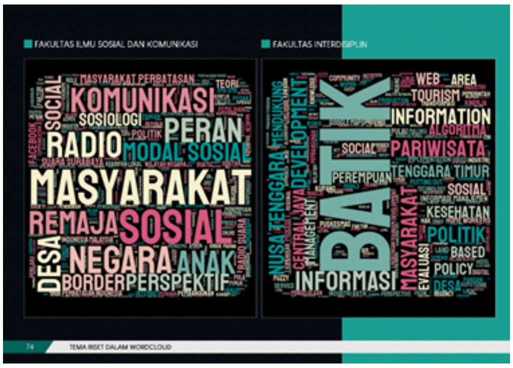


Figure 11. Word cloud for the research data of the Faculty of Social and Communication Sciences (left) and the Faculty of Interdisciplinary (right) with data up to July 2020.

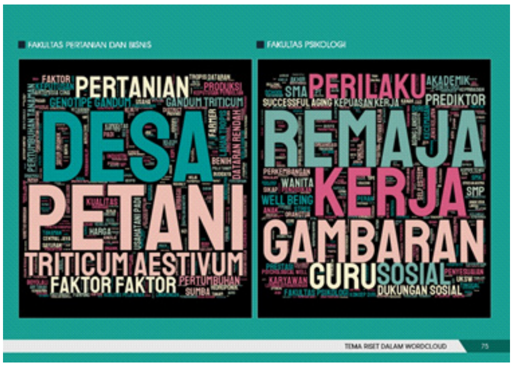


Figure 12. Word cloud for research data from the Faculty of Agriculture and Business (left) and the Faculty of Psychology (right) with data up to July 2020.

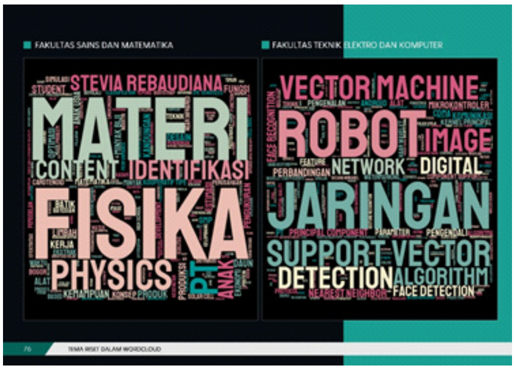


Figure 13. Word cloud for research data from the Faculty of Science and Mathematics (left) and the Faculty of Electrical and Computer Engineering (right) with data up to July 2020.

In the word cloud research that is in UKSW's flagship research, each faculty has different words that stand out. This means that each faculty has different research and results from one faculty to another.

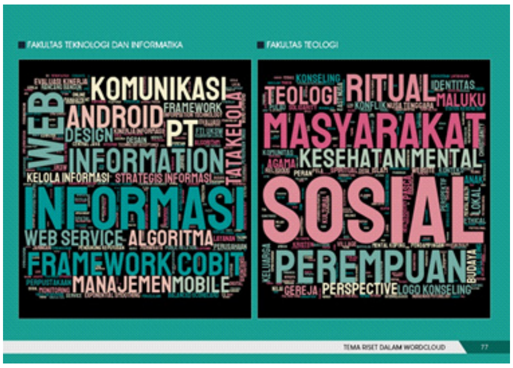


Figure 14. Word cloud for research data from the Faculty of Technology and Informatics (left) and the Faculty of Theology (right) with data up to July 2020.

At the Faculty of Language and Literature, for example, the most prominent words are "English", "Student", "Teacher", "Teaching", and "Music". This means that the Word cloud can show that these words are words with a frequency that stands out from the number of word repetitions in the research at the UKSW Language and Literature Faculty.

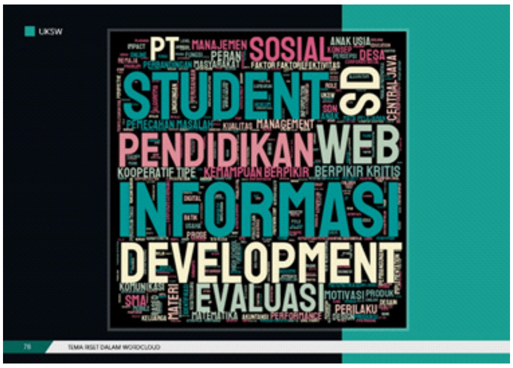


Figure 15. Word cloud for all SWCU research data with data up to July 2020.

It is also shown that the larger the letters, the more popular the topics/reports containing these sentences that are raised in every research in the faculty.

Then in the Faculty of Biology, the words that appear most often are "SMA", "Leaf", "SMP", "Tempe", and "Bacteria". So it can be said that the biology faculty often raises research that uses these words or even research with themes that raise these words such as Tempe or Bacteria. Next in the Faculty of Economics and Business, the words that often appear are "Accounting", "Company", "Corporate Social", "Bank", and "Social Responsibility". This shows that in the Faculty of Economics and Business the word in research or research that appears is around these words. And the most often used are the words Company and Accounting because the words that stand out the most and are the biggest are those two words. At the Faculty of Law, the words that appear most often are "Law", "International", and "law". This shows that in excellent research that is often carried out containing these words, it cannot be separated from the material at the Faculty of Law. In the Faculty of Social Sciences and Communication, the frequency of words that often appear is "Society", "Social", "State", "Communication" and "role" This means that the word frequency that is often used in research or research in this faculty is about those topics. In the Interdisciplinary faculty, the word that often appears is "Batik" because the word that stands out in the word Batik means that in this faculty often uses the word/research containing the word Batik. Then at the Faculty of Agriculture and Business, the frequency of words that often appear is the words "Village", "Farmers", "Agriculture", and "Triticum Aestivum". This shows that the research or research that is often carried out by the Faculty of agriculture and business is to raise the theme of the words that stand out in the Word cloud. Then in the Faculty of Psychology, the words that often appear are "Teenagers", "Work", "Images", "Teachers", and "behavior" this is evidenced by the presence of the most prominent words that have the largest size. This means that in the psychology faculty often raises themes or words that use these sentences. In the Faculty of Science and Mathematics, the frequency of words that often appear are the words “Material”, “Physics”, “Stevia rebaudiana”, and “Identification”. Similarly, this shows that the Faculty of Science and Mathematics often uses these words as research or research being carried out. Then at the Faculty of Electrical and Computer Engineering, the most prominent words are the words “Robot”, “Network”, “Digital”, “Support Vector”, and “Vector Machine”. This shows that the Faculty of Electrical and Computer engineering often uses words or themes on these. Then at the Faculty of Technology and Informatics, the frequency of words that often appear is the words “Information”, “Framework Cobit”, “WEB”, “Communication” and “Android”. Observing the Faculty of Theology, the words “Social”, “Society”, “Ritual”, “Mental Health”, and “Women” appeared most frequently. Finally, the studied is carried out for the whole university. The words that stand out are the words “Student”, “Information”, “Education”, “Development”, “WEB”, and “Evaluation”. This shows that at UKSW, these words are most frequently carried out in the researches done in the Faculty of Theology. The accuracy of the word cloud obtained is confirmed with existing data and communicated directly with the relevant parties. The results are expected as desired. However, it turns out that these six words do not appear dominantly from the titles of reputable articles where reputable journals are considered to have greater weight than those that are not reputable.

The explanation above is done by observing the results of the visual word cloud that appears where the researcher can search quickly because of the prominent appearance of the word in the observed word cloud. This is in accordance with the purpose of the urgency of research, namely to get information quickly on the characteristics of each study program or faculty through the word cloud without paying attention to the TF-IDF that is in each result considering the limited research time and it is necessary to immediately report the research results in November 2020 on related parties so that further management steps can be taken with the results of this research. In addition, other authors who work with word clouds also provide a similar analysis where the analysis is carried out by paying attention to the visual results of the word cloud that are displayed [16]. However, the TF-IDF algorithm can also be used for analysis even with the modified TF-IDF algorithm which weights have been shown to give better results in the case of word analysis of Chinese documents [17]. By comparing these results, it is hoped that in further research, weighting can be carried out on article titles with a higher reputation than journals with lower categories (for example those that are not accredited) so that the results of the word cloud can create a dominant visualization for article titles in reputable articles.

4. Conclusion

In this study, it was demonstrated about making a word cloud for Universitas Kristen Satya Wacana (UKSW) lecturer research data based on the data on lecturer article titles that were documented on Google Scholar. This was done because efforts to collect data on UKSW research results often encountered difficulties because the data were not documented in an integrated manner. In addition, to track lecturers’ research competencies, it cannot be done by tracing documents manually in order to be able to make global conclusions in a timely and continuous manner. The used method is the machine learning method using the word cloud generator in Python where the corpus is built based on the data on the titles of the lecturer articles on Google Scholar so that the corpus in built specifically for this purpose.

The obtained results from this study are the visualization of word cloud leading to find out more easily classification of the dominant topics researches done by lecturers in University in the period until July 2020. These visualizations of word clouds have been carried out based on IDs lecturers of google scholars in study programs, faculties, and at university. Meanwhile, the outstanding results of each lecturer indexed by Scopus or reputable journals have not been identified dominantly on the results of word cloud. Therefore, in further research, this research will be corrected by paying attention to this sorting so that reputable journals get the highest frequency and it appears in the word cloud more dominantly that articles in reputable journals get better ratings than unreputable. Likewise, articles in accredited journals will dominate the word cloud more clearly than articles in unaccredited journals.

Acknowledgement

This research was funded by a UKSW research center with internal research for the 2020/2021 fiscal year with a research entitled “UKSW Leading Research Analysis using Machine Learning and Google Scholar Data”.

Reference

[1] V. B. Kobayashi, S. T. Mol, H. A. Berkers, G. Kismihók, and D. N. Den Hartog, “Text Classification for Organizational Researchers: A Tutorial,” *Organ. Res. Methods*, vol. 21, no. 3, pp. 766–799, 2018, doi: 10.1177/1094428117719322.

[2] R. Kusumaningrum, S. Adhy, and Suryono, “WCLOUDVIZ: Word cloud visualization of Indonesian news articles classification based on Latent dirichlet allocation,” *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 16, no. 4, pp. 1752–1759, 2018, doi: 10.12928/TELKOMNIKA.v16i4.8194.

[3] C. An, H. Lim, D. W. Kim, J. H. Chang, Y. J. Choi, and S. W. Kim, “Machine learning prediction for mortality of patients diagnosed with COVID-19: a nationwide Korean cohort study,” *Sci. Rep.*, vol. 10, no. 1, pp. 1–11, 2020, doi: 10.1038/s41598-

020-75767-2.

[4] J. Beschi Raja, R. Anitha, R. Sujatha, V. Roopa, and S. Sam Peter, “Diabetics prediction using gradient boosted classifier,” *Int. J. Eng. Adv. Technol.*, vol. 9, no. 1, pp. 3181–3183, 2019, doi: 10.35940/ijeat.A9898.109119.

[5] H. Qian, “Big data Bayesian linear regression and variable selection by normal-inverse-gamma summation,” *Bayesian Anal.*, vol. 13, no. 4, pp. 1007–1031, 2018, doi: 10.1214/17-BA1083.

[6] Y. A. Sari, A. G. Hapsani, S. Adinugroho, L. Hakim, and S. Mutrofin, “Preprocessing of Skin Images and Feature Selection for Early Stage of Melanoma Detection using Color Feature Extraction,” *Int. J. Artif. Intell. Res.*, vol. 4, no. 2, p. 95, 2021, doi: 10.29099/ijair.v4i2.165.

[7] L. Demidova, E. Nikulchev, and Y. Sokolova, “Big Data Classification Using the SVM Classifiers with the Modified Particle Swarm Optimization and the SVM Ensembles,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 5, pp. 294–312, 2016, doi: 10.14569/ijacsa.2016.070541.

[8] D. Antons, E. Grünwald, P. Cichy, and T. O. Salge, “The application of text mining methods in innovation research: current state, evolution patterns, and development priorities,” *R D Manag.*, vol. 50, no. 3, pp. 329–351, 2020, doi: 10.1111/radm.12408.

[9] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, “Machine learning for email spam filtering: review, approaches and open research problems,” *Heliyon*, vol. 5, no. 6, 2019, doi: 10.1016/j.heliyon.2019.e01802.

[10] Y. Jin, “Development of Word Cloud Generator Software Based on Python,” in *Procedia Engineering*, 2017, vol. 174, pp. 788–792, doi: 10.1016/j.proeng.2017.01.223.

[11] G. Sazandrishvili, “Asset tokenization in plain English,” *J. Corp. Account. Financ.*, vol. 31, no. 2, pp. 68–73, 2020, doi: 10.1002/jcaf.22432.

[12] G. Astika, “Lemmatizing textbook corpus for learner dictionary of basic vocabulary,” *Indones. J. Appl. Linguist.*, vol. 7, no. 3, pp. 630–637, 2018, doi: 10.17509/ijal.v7i3.9813.

[13] Hartanto, “Text Mining Dan Sentimen Analisis Twitter Pada Gerakan Lgbt,” *Intuisi J. Psikol. Ilm.*, vol. 9, no. 1, pp. 18–25, 2017.

[14] N. K. Widyasanti, I. K. G. Darma Putra, and N. K. Dwi Rusjayanthi, “Seleksi Fitur Bobot Kata dengan Metode TFIDF untuk Ringkasan Bahasa Indonesia,” *J. Ilm. Merpati (Menara Penelit. Akad. Teknol. Informasi)*, p. 119, 2018, doi: 10.24843/jim.2018.v06.i02.p06.

[15] M. Umadevi, “Document Comparison Based on the Page Layout,” no. 1, pp. 2–6, 2020

[16] Y. Huang, Y. Wang, and F. Ye, “A Study of the application of word cloud visualization in college english teaching,” *Int. J. Inf. Educ. Technol.*, vol. 9, no. 2, pp. 119–122, 2019, doi: 10.18178/ijiet.2019.9.2.1185.

[17] J. Chen, C. Chen, and Y. Liang, “Optimized TF-IDF Algorithm with the Adaptive Weight of Position of Word,” 2016, vol. 133, pp. 114–117, doi: 10.2991/aiie-16.2016.28.

khazanah informatika

Jurnal Ilmu Komputer dan Informatika

Load Balancing Server and Homomorphic Encryption in Internet of Things

Muhammad Hafiz Amrullah, Favian Dewanta[*], Sussi
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
*favian@telkomuniversity.ac.id

Abstract-User demand for Internet of Things (IoT) services has been increasing. The growing number of user demand can lead to an escalation of server workloads and threat of critical data theft. Consequently, a system is necessary to balance the server load where the data is protected with encryption. In this study, we designed a system to share server workloads using load balancing methods. The load balancing technique uses open-source web server software. The system is equipped with data security using a homomorphic encryption algorithm from AES on the sender's side. The system embeds in an IoT telemedicine apparatus. During testing, we analyze the error requests that arrive at each server for the HTTP GET and POST methods. We also evaluate the speed of data encryption and decryption. The results showed that server load balancing reduces the number of error requests for the GET method by 97%. Meanwhile, the number of error requests for the POST method decreases by 66.75%. Observations reveal that the average homomorphic encryption speed, computation time, and decryption time are 15.66 ms, 764.18 µs, and 362.49 µs, respectively.

Keywords*:* load balancing, servers, requests, homomorphic encryption, AES algorithm

Article info: *submitted February 3, 2021, revised April 6, 2021, accepted May 26, 2021*

1. Introduction

The current industrial development encourages the development and application of the Internet of Things (IoT). The main objective of IoT technology is to enable connected devices to communicate with each other, exchange data, store data, and perform computing complying with the user requirement. However, there are several obstacles in IoT implementation, one of which is the server load and data security on the server and database. When the number of IoT users requesting service increases and the server cannot handle the requests, the server will receive too many requests that may cause the service to fail to respond as expected [1].

To overcome these problems and improve server performance, server load balancing and homomorphic encryption systems can be implemented as a solution in terms of uniform distribution of service loads and data protection with fast computing. The system implementation expectedly improves the reliability and security of the IoT system. Server load balancing works by considering the capacity of each server and distributing workloads to several servers, which may reduce failures on the servers [2]. Homomorphic encryption is a cryptographic algorithm that allows (arithmetic) computing on the ciphertext directly. It avoids the encryption process on the plain text, which forces the decryption process, which prolongs the steps. Homomorphic encryption result is the same with encryption to plaintext [3]. It allows safe data storage in the database in the form of ciphertext, and it achieves a faster processing time compared to the use of regular cryptographic algorithms.

In this study, the server design adopts an open-source webserver to implement a load balancing system and apply homomorphic encryption on a web server that runs on an IoT-based telemedicine system. Testing parameters include the error request received by the server and the length of time the homomorphic encryption process runs on the server.

2. Basic theory

a. Load Balancing

Load balancing is a technology to divide the traffic load evenly on two or more connection lines in a balanced way so that traffic does not experience congestion and can run optimally. The target of load balancing is to maximize throughput, reduce response time, and anticipate overload on one connection line [4]. If the service users on the server continue to grow and exceed the capacity available in the network traffic path, the load balancer will distribute the load from users evenly to all available servers. In addition, load balancers can also evenly distribute loads of CPU, hard disk, RAM, and other computing resources to get the best performance from the server.

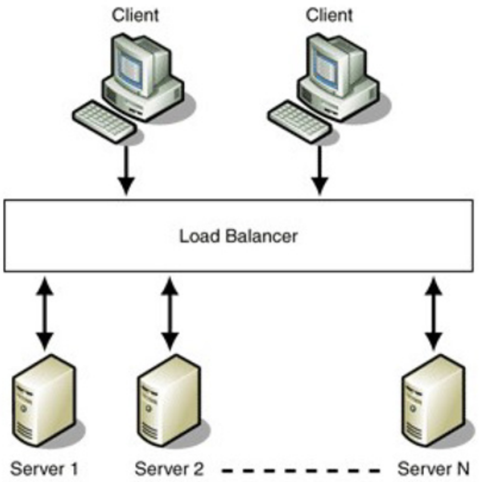


Figure 1. Typical load balancer configuration [4].

In this study, we implement a load balancer using open-source software called NGINX that uses the Round Robin algorithm which has its server or is separate from other servers. The Round Robin algorithm is the simplest algorithm and the most widely used algorithm for load balancer devices. The Round Robin algorithm works by distributing the load sequentially from one server to another. The basic concept of the Round Robin algorithm uses time-sharing, which simply processes the queue (traffic or computation) in turn [5].

b. Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES) is an encryption algorithm with a symmetric key exchange and applies a block cipher system that has a block length of 128 bits [6]. In cryptography, there are terms plain text (plaintext) and ciphertext (ciphertext). Plaintext is the initial information or data before the information or data is encrypted, and ciphertext is information or data from encrypted plain text. AES encryption uses a different number of round keys for each type of block size, namely a block length of 128 bits is 10 rounds, a block length of 192 bits is 12 rounds, and a block length of 256 bits is 14 rounds [7].

The type of AES block cipher used in this study is AES Cipher Block Chaining (CBC) 256 bits as illustrated in Figure 2. The advantage is that if the information or data has the same plaintext, the encrypted information or data cannot be repeated with the same encryption. This is due to the use of an Initialization Vector (IV) which has a different and random value for each data encryption.

CBC is the operating mode of the block cipher whose IV length is the same as each plaintext block. The initial process of encryption is to XOR plaintext with IV, and then generate encrypted data (ciphertext) for plaintext blocks. Then the resulting ciphertext will be used as IV again in the next block. In this way, each ciphertext generated will depend on each ciphertext in the previous block, so that each encrypted data becomes unique [8].

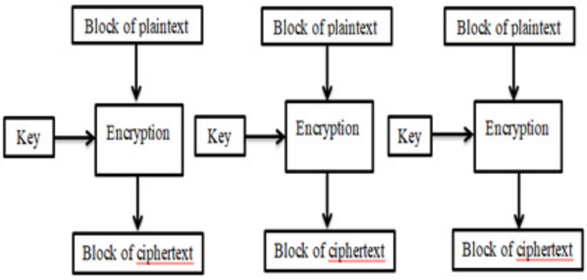


Figure 2. CBC mode of operation [9].

c. Homomorphic encryption

Homomorphic encryption is a cryptographic algorithm that makes it possible to compute encrypted data without decrypting the data directly as the concept is described briefly in Figure 3. . Homomorphic encryption uses an encryption function with addition (addition) and or multiplication (multiplication) operations on encrypted data [10].

There are two types of homomorphic encryption, namely Partially Homomorphic Encryption (PHE) and Fully Homomorphic Encryption (FHE). PHE is homomorphic encryption, which allows certain types of operations to be used on the ciphertext. While FHE is a homomorphic encryption, which allows two operations, namely addition and multiplication of ciphertext.

In this research, a homomorphic FHE encryption process is applied from AES encrypted data which is used to calculate the average value of the data sent from the sender. This homomorphic encryption process uses the help of a python library called Pyfhel.

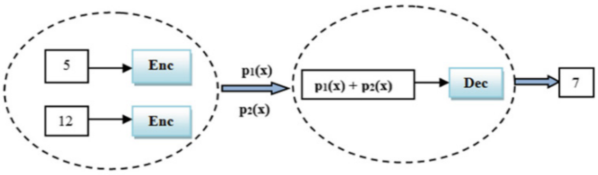


Figure 3. Homomorphic encryption concept [10].

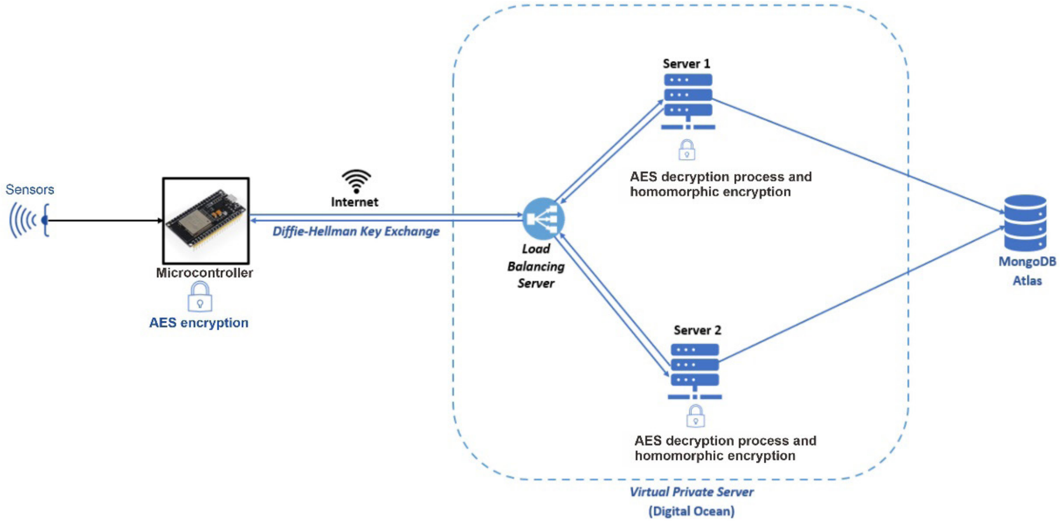

Sensors
Microcontroller
AES encryption
Internet
Diffie-Hellman Key Exchange
Load Balancing Server
Server 1
AES decryption process and homomorphic encryption
Server 2
AES decryption process and homomorphic encryption
MongoDB Atlas
Virtual Private Server (Digital Ocean)

Figure 4. Design of the system created.

3. Method

a. System planning

At this stage, the author designs a system from sensors to the cloud server and its database as shown in Figure 4. The author also analyzes the communication method between the right elements in the system to be used in solving problems based on information from datasheets, tutorials, and other sources. others available.

Figure 4 shows the overall design of the telemedicine system. The input data comes from the sensor. The data sent by the microcontroller is the ciphertext data from AES, the shared key to calculate the secret key, and the IV generated from the microcontroller. Before encryption, the microcontroller and server use the Diffie-Hellman algorithm to exchange shared keys to calculate the symmetric secret-key value to perform the AES encryption process and generate the ciphertext. The ciphertext data is passed to the VPS via the created web server API. The ciphertext data first passes through the load balancer of the webserver which distributes the data traffic to one of the two servers used. After that decrypt the AES ciphertext data received by the server. After the ciphertext is decrypted, the plaintext results are then encrypted and computed using a homomorphic encryption algorithm to calculate the average value per 100 data received by server 1 and server 2, and the results of the average are decrypted again using a homomorphic algorithm. Based on the average value of the plaintext data, AES encryption is performed again using the received key and IV. After the AES encryption process is complete, the ciphertext data will be saved to the MongoDB Atlas database.

Figure 5 is a system flow diagram starting with the microcontroller performing the Diffie-Hellman key exchange process. First, the process of making shared keys X and IV, then the microcontroller will get the shared key Y from the server and then the shared keys X and IV are sent to the server. The microcontroller will generate the secret key K from the shared key Y computation, and then perform hashing to get the 256-bit key. After that, the AES algorithm is used to encrypt the data and generate the ciphertext. The AES ciphertext is sent to the cloud server. While on the cloud server, the data first arrives at the load balancer which is created using the Round Robin algorithm with the initial process of scheduling all user requests that enter the load balancer until all user requests are scheduled, the load balancer will distribute user requests to one of the servers. If the selected web server is overloaded, then the data requested by the user is reset to be sent to another web server. After the requested data is received by one of the servers, the request data in the form of ciphertext is decrypted and computed using homomorphic encryption. Once the computation is complete, the data will be passed to the MongoDB Atlas database for storage.

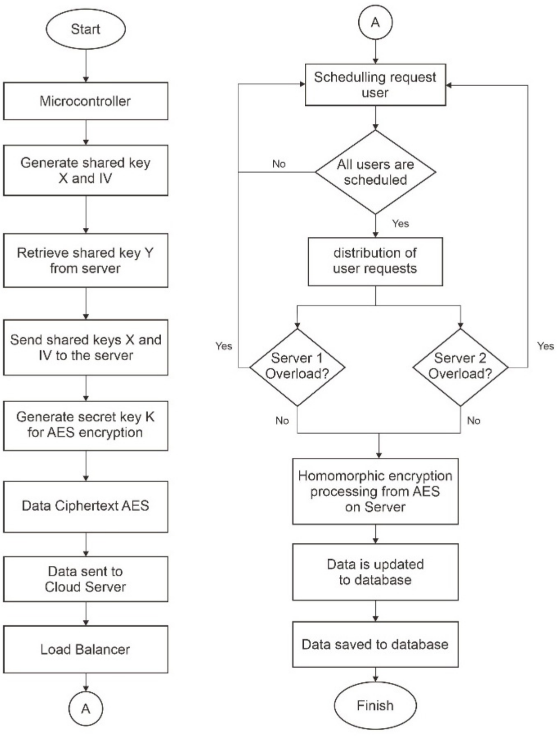
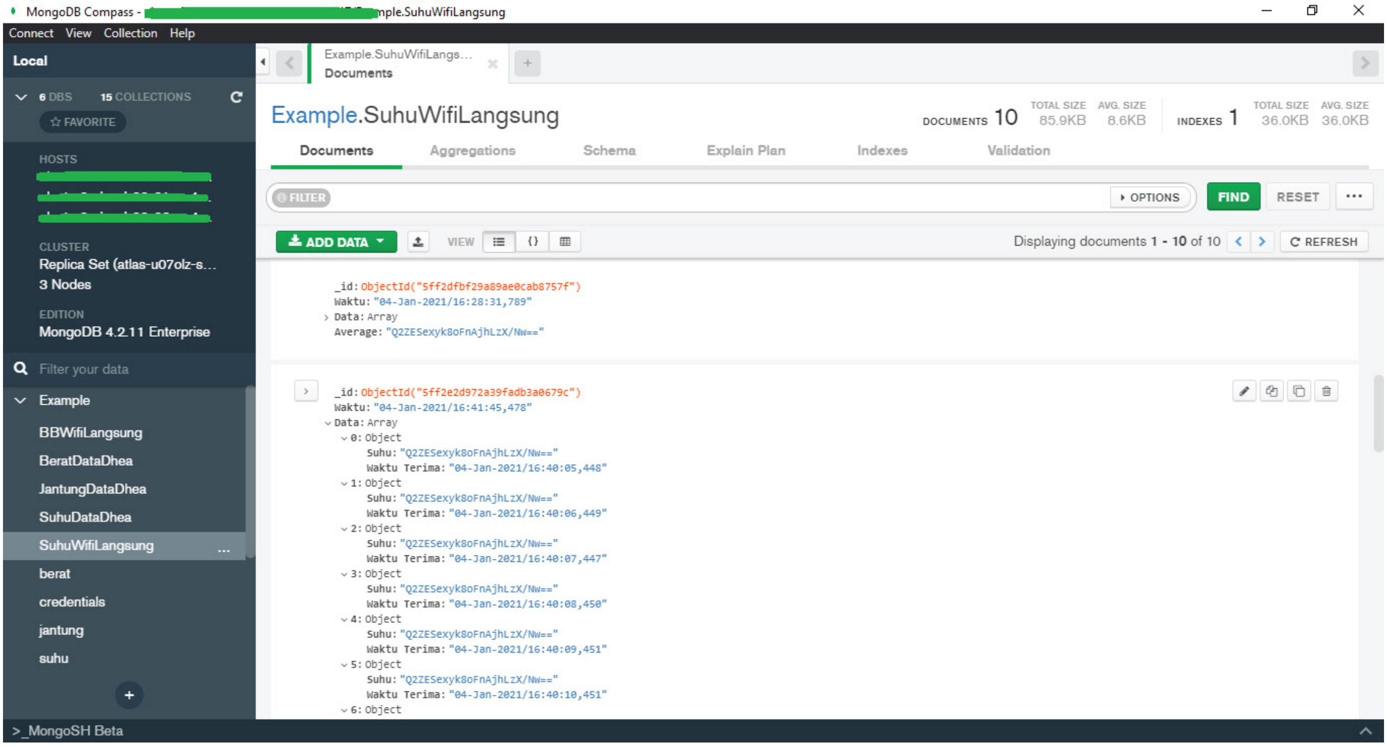

Start
Microcontroller
Generate shared key X and IV
Retrieve shared key Y from server
Send shared keys X and IV to the server
Generate secret key K for AES encryption
Data Ciphertext AES
Data sent to Cloud Server
Load Balancer
A
A
Schedulling request user
All users are scheduled
No
Yes
distribution of user requests
Server 1 Overload?
Server 2 Overload?
Yes
Yes
No
No
Homomorphic encryption processing from AES on Server
Data is updated to database
Data saved to database
Finish

Figure 5. Process diagram flow of the system.



MongoDB Compass - [redacted]mple.SuhuWifiLangsung

[redacted]

```
_id: ObjectId("5ff2dfbf29a89ae0cab8757f")
Waktu: "04-Jan-2021/16:28:31,789"
> Data: Array
Average: "Q2ZESexyk8oFnAjhLzX/Nw=="

_id: ObjectId("5ff2e2d972a39fadb3a0679c")
Waktu: "04-Jan-2021/16:41:45,478"
v Data: Array
  v 0: Object
      Suhu: "Q2ZESexyk8oFnAjhLzX/Nw=="
      Waktu Terima: "04-Jan-2021/16:40:05,448"
  v 1: Object
      Suhu: "Q2ZESexyk8oFnAjhLzX/Nw=="
      Waktu Terima: "04-Jan-2021/16:40:06,449"
  v 2: Object
      Suhu: "Q2ZESexyk8oFnAjhLzX/Nw=="
      Waktu Terima: "04-Jan-2021/16:40:07,447"
  v 3: Object
      Suhu: "Q2ZESexyk8oFnAjhLzX/Nw=="
      Waktu Terima: "04-Jan-2021/16:40:08,450"
  v 4: Object
      Suhu: "Q2ZESexyk8oFnAjhLzX/Nw=="
      Waktu Terima: "04-Jan-2021/16:40:09,451"
  v 5: Object
      Suhu: "Q2ZESexyk8oFnAjhLzX/Nw=="
      Waktu Terima: "04-Jan-2021/16:40:10,451"
  v 6: Object
```

Figure 6. MongoDB database view.

Figure 6 is a display of ciphertext data storage in MongoDB. The encrypted temperature data will be stored in an array variable named "Data" as many as 100 encrypted data and from the 100 data the average value is calculated using homomorphic encryption which is stored in the "Average" variable in an encrypted fixed form.

b. Implementation

At this stage, the author makes the system after the previous design has been completed.

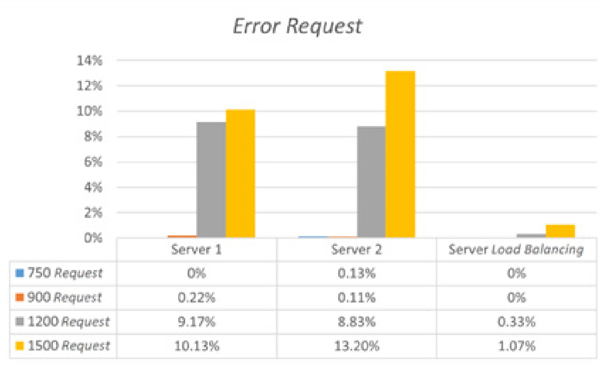
c. Testing and Analysis Phase

At this stage, the system is tested to observe the required data. The test is carried out by measuring QoS and obtaining data during the homomorphic encryption and decryption process.

The test is carried out using Apache Jmeter software, where the testing scheme for error request parameters is carried out by sending a different number of requests, namely 750, 900, 1200 and 1500 requests where all requests are sent within 10 seconds. To test AES processing speed and homomorphic processing, each experiment was performed 5 times by sending 100 requests within 100 seconds.

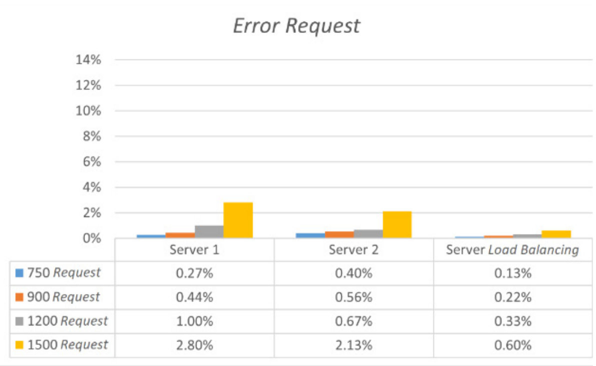
4. Result

Figure 7 and Figure 8 are the results of testing error requests received by the server for the GET and POST methods based on the number of different requests (ie 750, 900, 1200, and 1500 requests).



	Server 1	Server 2	Server Load Balancing
750 Request	0%	0.13%	0%
900 Request	0.22%	0.11%	0%
1200 Request	9.17%	8.83%	0.33%
1500 Request	10.13%	13.20%	1.07%

Figure 7. GET method request error chart.

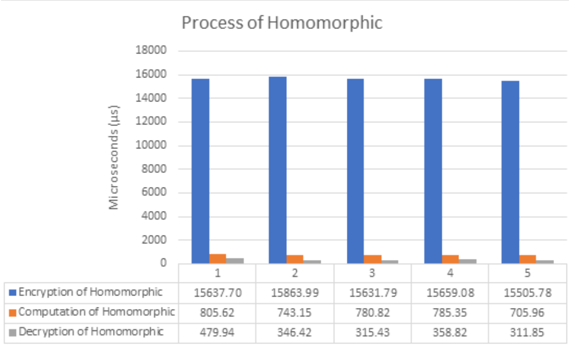


	Server 1	Server 2	Server Load Balancing
750 Request	0.27%	0.40%	0.13%
900 Request	0.44%	0.56%	0.22%
1200 Request	1.00%	0.67%	0.33%
1500 Request	2.80%	2.13%	0.60%

Figure 8. POST method request error chart.

The result of the error request test on the GET method shows that the error request results between server

1 and server 2 are almost the same. However, when using server load balancing techniques, the number of error requests received is reduced by an average of about 97%. At the same time, the results of the POST method error request testing show that the error request results between server 1 and server 2 also show almost the same results. However, on the load balancing server used, the number of error requests received was reduced by an average of about 66.75%. This proves that the load balancing server implemented for the IoT telemedicine system is operating properly.



	1	2	3	4	5
■ Encryption of Homomorphic	15637.70	15863.99	15631.79	15659.08	15505.78
■ Computation of Homomorphic	805.62	743.15	780.82	785.35	705.96
■ Decryption of Homomorphic	479.94	346.42	315.43	358.82	311.85

Figure 9. Homomorphic process time chart.

Figure 9 is the result of testing the time of the homomorphic encryption process which includes 3 processes, namely encryption, computation, and homomorphic decryption. When testing the speed of the homomorphic encryption and decryption process, the encryption and decryption process was carried out in 5 trials by sending 100 data to the server each time. The results of the encryption processing time show that the average homomorphic encryption process time is 15.66 milliseconds, the homomorphic computation time average is 764.18 microseconds (0.76 milliseconds), and the homomorphic decryption time average is 362.49 microseconds (0.36 milliseconds). The results of the computational speed test on the homomorphic cryptography algorithm show that the encryption process takes the most time because the encryption process in the algorithm uses a lot of computational overhead in processing plaintext data to ciphertext. This reason is sometimes a practical consideration not to apply homomorphic encryption to databases that require real-time data computation.

5. Conclusion

Server load balancing can reduce error requests received by the server. The test results show that error requests for the GET method are reduced by 97%, while error requests for the POST method are reduced by 66.75%. Homomorphic encryption can be applied to the server to calculate the average value of the data received by the server on the IoT system. It proves that homomorphic encryption runs well without errors. The average speed of homomorphic encryption, computation, and decryption is 15.66 ms, 764.18 s (0.76 ms), and 362.49 s (0.36 ms), respectively, which suggests that encryption works as expected.

Reference

[1] S. D. Riskiono and D. Pasha, “Analisis Metode Load Balancing Dalam Meningkatkan Kinerja Website E-Learning,” J. Teknoinfo, vol. 14, no. 1, p. 22, 2020, doi: 10.33365/jti.v14i1.466.

[2] S. D. Riskiono, “Implementasi Metode Load Balancing Dalam Mendukung Sistem Kluster Server,” pp. 455–460, 2018, doi: 10.31227/osf.io/9vuzx.

[3] Q. Wang, D. Zhou, and Y. Li, “Secure Outsourced Calculations with Homomorphic Encryption,” Adv. Comput. An Int. J., vol. 9, no. 6, pp. 01–14, 2018, doi: 10.5121/acij.2018.9601.

[4] A. Rahmatulloh and F. MSN, “Implementasi Load Balancing Web Server menggunakan Haproxy dan Sinkronisasi File pada Sistem Informasi Akademik Universitas Siliwangi,” J. Nas. Teknol. dan Sist. Inf., vol. 3, no. 2, pp. 241–248, 2017, doi: 10.25077/teknosi.v3i2.2017.241-248.

[5] F. Apriliansyah, I. Fitri, and A. Iskandar, “Implementasi Load Balancing Pada Web Server Menggunakan Nginx,” J. Teknol. dan Manaj. Inform., vol. 6, no. 1, 2020, doi: 10.26905/jtmi.v6i1.3792.

[6] X. W. Wu, E. H. Yang, and J. Wang, “Lightweight security protocols for the Internet of Things,” IEEE Int. Symp. Pers. Indoor Mob. Radio Commun. PIMRC, vol. 2017-Octob, pp. 1–7, 2018, doi: 10.1109/PIMRC.2017.8292779.

[7] B. K. S. Rajaram and N. Krishna Prakash, “Secure mqtt using aes for smart homes in iot network,” Int. J. Innov. Technol. Explor. Eng., vol. 8, no. 5s, pp. 483–485, 2019.

[8] A. M. Al Naamany, A. Al Shidhani, and H. Bourdoucen, “IEEE 802 . 11 Wireless LAN Security Overview,” Ijcsns, vol. 6, no. 5, pp. 138–156, 2006.

[9] M. E. Hameed, M. M. Ibrahim, N. A. Manap, and M. L. Attiah, “Comparative study of several operation modes of AES algorithm for encryption ECG biomedical signal,” Int. J. Electr. Comput. Eng., vol. 9, no. 6, pp. 4850–4859, 2019, doi: 10.11591/ijece.v9i6.pp4850-4859.

[10] Y. Alkady, F. Farouk, and R. Rizk, “Fully Homomorphic Encryption with AES in Cloud Computing Security,” Adv. Intell. Syst. Comput., vol. 845, pp. 370–382, 2019, doi: 10.1007/978-3-319-99010-1_34.

khazanah informatika

Jurnal Ilmu Komputer dan Informatika

Location Selection Based on Surrounding Facilities in Google Maps using Sort Filter Skyline Algorithm

Annisa*, Salsa Khairina
Department of Computer Science
IPB University
Bogor, Indonesia
*annisa@apps.ipb.ac.id

Abstract-Selecting a good location is an essential task in many location-based applications. Intuitively, a place is better than another if there are many good facilities around it. The most popular location selection platform today is Google Maps. Unfortunately, Google Maps has not provided the location selection based on the number of surrounding facilities. Assume a situation when a college student wants to let a house near his campus. Besides the distance from the campus, the student certainly will consider amenities surrounding it, such as food courts, supermarkets, health clinics, and places of worship. The rent house will become a better choice if there are more of these facilities around. Skyline query is a well-known method to select interesting desirable objects. We applied the Sort Filter Skyline (SFS) Algorithm on Google Maps to get a small number of attractive locations based on the number of nearby facilities. This study has succeeded in developing a web-based application that facilitates Google Maps users to search for places based on the figure of surrounding facilities. The time required to do a location search using SFS in Google Maps will increase with the number of facility types considered by the user.

Keywords: location selection, skyline query, sort filter skyline, surrounding facilities

Article info: *submitted December 17, 2020, revised April 5, 2021, accepted April 21, 2021*

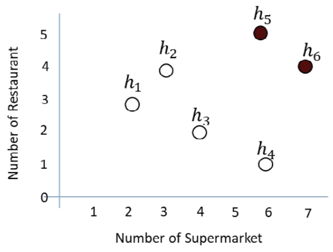
1. Introduction

Selecting a good location is an important task in many location-based applications. Intuitively one location is better than another if there are many good facilities/ objects around it. Nowadays, there is an increasing need to take surrounding facilities into account when selecting a location. Chang et al. [1] explained when someone goes to a place for business or leisure, choosing the best hotel becomes very important. Syafrianto [2] and Popovic et al. [3] explained that hotel selection is strongly influenced by the goals and needs of visitors, not only in the form of hotel facilities, but also geographical surrounding and public facilities around the hotel. Another real world example is a situation when a college student wants to rent a house near his/her campus. Besides considering the distance from the rent house to the campus, the student certainly will also consider what facilities are available around the house, such as places to eat, supermarkets, health clinics, and also places of worship. The rent house will be considered as better option if there are more of these facilities around.

Skyline query [4] is a widely known method for selecting small number of interesting objects. Interesting objects, also known as skyline objects, are non-dominated objects in d-dimensional database. Borzsonyi et al. [4] defined that an object is said to dominate another object if it is equally good in all dimensions and better in at least one dimension. Figure 1 illustrates the skyline query problem. Consider a college student wants to rent a house. H1 to H6 are the houses for rent. Table in Figure 1 (a) shows the number of supermarkets and restaurants surrounding each house. Using skyline query algorithm, user can get the list of interesting rent houses based on the number of supermarkets and restaurants surround them. In Fig. 1 (b), H5 and H6 are skyline objects. H1, H2, H3, and H4 are dominated by H5 and H6, because H5 and H6 has more surrounding supermarkets and restaurants. H5 and H6 do not dominate each other because H5 has a greater number of surrounding restaurants but a lower number of surrounding supermarkets than H6 and vice versa. Using skyline query we can suggest H5 and H6 to the college student as options for renting a house. He/she can choose H5 if his/her preference is more surrounding restaurants, otherwise he/she can choose H6. There are many skyline query algorithms, including [5, 6, 7, 8]. The skyline method has also been used in location and route selection such as in [9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20].

House	Number of Supermarket	Number of Restaurant
H1	2	3
H2	3	4
H3	4	2
H4	6	1
H5	6	5
H6	7	4

(a)



(b)

Figure 1. Illustration of skyline query problem

In [16], Arefin et al. uses skyline queries to address the problem of site selection by considering the type and number of surrounding facilities. Figure 2 shows an illustration of location selection problem in Arefin et al. [16]. Let us consider problem of the college student in finding rent house above.

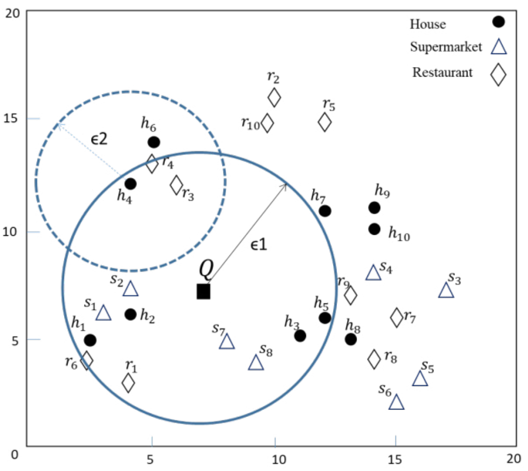


Figure 2. Illustration of location selection problem in Arefin et al. [15]

At first step, user gives the reference point (Q) and a radius distance (ϵ1). This Q reference point is generally a popular place in an area. In this example, campus is the reference point. Based on ϵ1 distance from Q, spatial objects of the target facility are selected. The houses (h) in Figure 2 are the examples of target facility. For the next step, we only consider rent houses within distance ϵ1 from Q. In the second step, the user determines the facility types that will be considered around the target (candidate) objects (houses within ϵ1 distance from Q). In the third step, if user considers n-surrounding facility types, then the number of facility types 1 to n at radius ϵ2 of each target object will be counted. In this example, user considers two types of facilities, restaurant (r)and supermarket (s). Thus the number of restaurant and supermarket within radius ϵ2 from each target objects is calculated and used as an attribute for each target object, to be combined with non-spatial information such as ratings, prices, and so on. Arefin et al. uses a variant of the R-tree index structure called aR-tree [21] to store spatial and non-spatial information from each facility types. In the last step, skyline query algorithm is performed like in Figure 1 to select skyline objects from the previous target objects. Arefin et al. applied Sort Filter Skyline (SFS), an algorithm introduced by Chomicki et al. [22] to find skyline objects. SFS is an enhancement of the naïve skyline algorithm, namely Block Nested Loops (BNL). SFS uses entropy function value to presort the dataset to reduce domination comparison.

Although the type of query like in [16] is very important in location selection application, unfortunately it has not been widely applied in our society. For this reason, this research aims to develop a web-based application so that users can run query in [16] on the most popular location selection platform, Google Maps. Many researches have been using Google Maps [23, 24, 25, 26, 27], however based on our knowledge, no research has been considering number and types of surrounding facilities for location selection in Google Maps.

Generally, Google Maps query, like Place Search and Nearby Search allow us to search for place information in specified area using a variety of categories. It returns a list of place location along with the summary information about each place, but never consider number of surrounding facilities for the selection. In this paper we developed a web-based application using SFS algorithm and Google Maps API so it can facilitate Google Maps users to search for location based on the number of surrounding facilities.

The remainder of this paper is organized as follows. In section II we briefly present location selection based on surrounding facilities problem in [16]. Section III provides our research methodology in detail. In section IV we conducted some experiments and explained them briefly in results and discussions. Finally, we put our conclusion in Section V.

2. Sort Filter Skyline (SFS) Algorithm

Skyline query has been widely used for location selection. Kodama et al. [28] and Wong et al. [29] introduced a framework for skyline query considering surrounding facilities using one type of facility. Arefin et al. [16] consider more types of facilities in location selection. In their research Arefin et al. explained that there are four calculation steps to obtain locations that take into consideration surrounding facilities according to user preference as we mentioned in the previous section. In the final step, the SFS algorithm is used to get skyline objects from a number of existing candidate objects.

Sort Filter Skyline (SFS) is a skyline query algorithm introduced by Chomicki et al. [22]. SFS is an enhancement of the naïve skyline algorithm, namely Block Nested Loops (BNL). Block Nested Loops (BNL) is an algorithm to read input data and compares each input with existing objects in memory. BNL does not use indexes and sorting. At first run, the first data directly enters the memory because there are no other objects in the memory. Subsequently until all input data is read, if the input object is dominated by at least one object in memory then the input object is eliminated. If the input object is not dominated by one or more objects in memory, then the input object is nominated as skyline and any object that is dominated by the input object is removed from memory.

SFS is the development of the BNL algorithm by sorting data according to its entropy function values. Entropy formula is:

$$E(t) = \sum \ln(t[h_i]+1) \qquad (1)$$

where is normalization of the candidate object attribute values (like rating and number of surrounding facilities). The entropy function above is always a monotone scoring function [8]. Based on Kalyvas et al. [30], the equation produces the most effective filtering of objects during skyline calculation. Intuitively, the smaller the entropy value of an object, the less likely it is to dominate.

Chomicki et al. [22] describes that an object in data which has a high entropy value can eliminate more objects away because it is guaranteed to dominate other objects. Therefore, processing sorted data has the advantage that no object in the data can be dominated by any object that enters afterwards. Thus, the data sorted by the value of the entropy function can eliminate the dominated object quickly.

The entropy function value is used to help filtering skyline objects by sorting data. SFS algorithm processes data that has been sorted using the entropy function. The object that has the largest entropy value is the skyline so it will go straight into memory because the object dominates other objects. After that, the next object is compared to the skyline object in memory. If the object is dominated by an object in memory then the object is eliminated because it is not a skyline object. On the other hand, if the object is not dominated by any object in memory then the object is saved to memory because it is a skyline object. Therefore, it is enough to compare an input object with skyline objects in memory without having to compare with all objects in the data. This happens because objects that have been previously eliminated are already dominated by skyline objects in memory, so it is enough to compare the input object with objects already in memory.

3. Methods

The data used in this research is Point of Interest (POI) data from Google Maps. POI data consists of spatial and non-spatial information. The spatial information is in the form of POI locations (latitude and longitude), while non-spatial information is in the form of POI type and POI rating. We use radius (near-by) feature that is already available in the Google Maps API to implement radius ϵ1 and ϵ2.

This study consisted of 5 stages. The first step is to determine the input, then proceed with making a data collection module. After the data obtained, then the data is processed by the Sort Filter Skyline (SFS) module, then the system is implemented, after that the system is tested, and finally conducting experiments with several scenarios. We used sample data to simplify the explanation of the method that we use in this research.

a. Determining Input Data

In this stage, we determined the input data needed to perform skyline operations. User input are spatial and non-spatial information. The information expected from the user are: the reference location that will become the reference point for searching spatial objects (*Q*), the type of object desired (type of target objects), the maximum radius from the reference location (ϵ1), the type of surrounding facilities, the maximum radius of the facilities from target objects (ϵ2), and the minimum rating from the surrounding facilities.

b. Creating Data Collection Module

The data collection module utilized two types of Google Maps' Place API. First we used Find Place to get geometry information (latitude and longitude) of the reference point, using reference point's place name as parameter. For example, if user input IPB University as the location of the reference point, the API returns geometry information of IPB University, which are latitude: -6.56636555 and longitude: 106.72148035. This geometry information is used as parameter to search for target objects surrounding the location of the reference point

Google Maps API Nearby Search is then used to get candidate objects and facilities surrounding the reference point, along with the required non-spatial information, which is rating. The parameters used are the geometry information of the search location point, the type of spatial object around the location of the reference point, the surrounding facilities (restaurants, ATMs, etc.), and the maximum radius. Table 1 shows candidate objects of place for rent within 0.7 km from IPB University: Landhius IPB Guest House, IPB International Dormitory, Amarilis Guest House, Al-Quds Boarding House, Arif Dormitory, and Dramaga Village Boarding House. After obtaining a candidate (target) object, this API is also used to find the number of facilities around the candidate object using the radius information from user. Maximum number of data obtained from requests for one type of POI is 60 data. User can input minimum rating of facility type, or can choose "None" in the system to not consider rating information from the facilities around the candidate object. This data collection module is run online to get data from Google Maps.

The results of this module are candidate objects data along with rating and number of facilities around them. Table 1 is an example of the results from the data collection module. Some candidate objects do not have rating info, so the rating value is displayed as 0.

Table 1. Example of data collection module results

Candidate Object	Rating	Restaurant-count	ATM-count
Landhius IPB Guest House	4.3	9	2
IPB International Dormitory	4.5	10	2
Wisma Amarilis	4.3	0	1
Arif Dormitory	0	9	1
Al-Quds Boarding House	0	9	2
Dramaga Village Boarding House	4.6	9	2

c. Creating a Sort Filter Skyline (SFS) Module

SFS algorithm is implemented using Python. Input data for SFS module is the result of the data collection module, as displayed in Table 1. Within SFS module, the input data is then sorted based on the highest to lowest entropy function values of the data calculated using (1), which results is presented in Table 2. Because the maximum value of the candidate object is 60, the candidate object attribute value is normalized by multiplying each candidate object by 0.001 so that the candidate object entropy value is between 0 and 1.

Table 2. Candidate location after sorted by entropy value

Candidate Object	Rating	Restaurant-count	ATM-count	Entropy Value
IPB International Dormitory	4.5	10	2	0. 016365
Dramaga Village Boarding House	4.6	9	2	0. 015480
Landhius IPB Guest House	4.3	9	2	0.015184
Al-Quds Boarding House	0	9	2	0.010940
Arif Dormitory	0	9	1	0.009950
Amarilis Guest House	4.3	0	1	0.005286

Subsequently, the objects that are dominated by other objects are removed. Thus, the obtained skyline objects are Dramaga Village Boarding House and IPB International Dormitory. Dramaga Village Boarding House is skyline because it has the highest rating, while IPB International Dormitory has the highest number of surrounding restaurants.

SFS algorithm is able to return skyline objects that consider several types of surrounding facilities from a candidate object. Currently, Google Maps is only able to consider just one type of facility from a location.

d. System Implementation

System implementation stage is carried out with the design and development of SSQ in web-based application. Web design phase is started by creating activity diagram that illustrates the flow of the system, as shown in Figure 3. Then simple mockups is made for developers to build the system.

Next, a web application prototype was developed based on activity diagrams and system requirements. The Google Maps API used is the same as in the data collection module, which is the Place API. The parameters required by the API are obtained from user input. Figure 4 and 5 is the result of the web interface on the location search feature based on the type and number of surrounding facilities which consist of the location search form page and the result recommendation location page.

The system processes Google Maps data to get location recommendations based on user input. Users are asked to fill in their preferences in a form, then the system provides a list of recommended locations according to the preferences given by the user.

Figure 4 is a form page that the user must complete. The reference location is a reference point for users to search for spatial objects. The type of object searched is the type of spatial object desired by the user around the reference point. The maximum object radius is the maximum radius from the reference point to find the desired object. Types of facilities around the object are the facilities preferred user wants around the spatial object. The maximum radius of the surrounding facility is the maximum radius of the facility of each object in unit of meter and the minimum rating of the surrounding facility is the minimum rating desired by the user for each facility type.

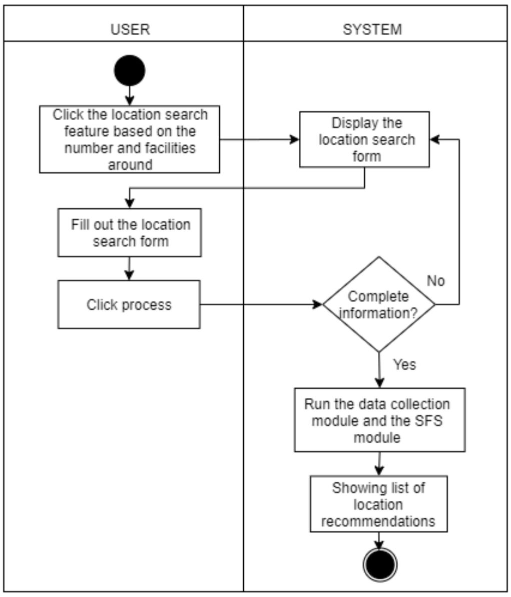
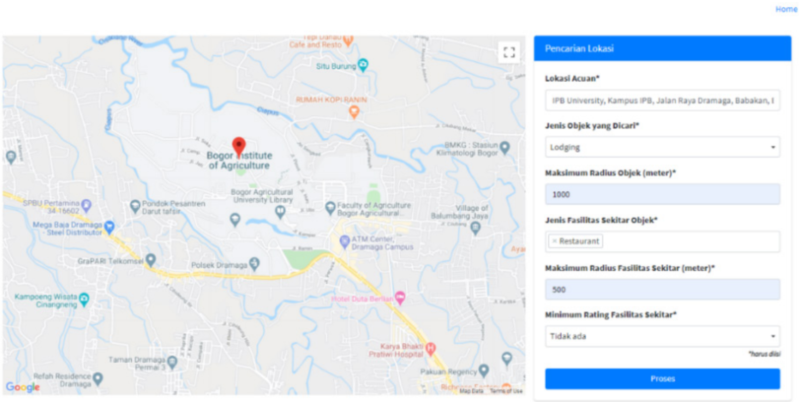

USER
SYSTEM
Click the location search feature based on the number and facilities around
Display the location search form
Fill out the location search form
Click process
Complete information?
No
Yes
Run the data collection module and the SFS module
Showing list of location recommendations

Figure 3. Activity Diagram

Figure 5 is a result page where there are red and green icons. The green icon is the location / reference point. The red icon is the result of the skyline which is the recommended locations for users based on the given preference.



Pencarian Lokasi

Lokasi Acuan*: IPB University, Kampus IPB, Jalan Raya Dramaga, Babakan, I

Jenis Objek yang Dicari*: Lodging

Maksimum Radius Objek (meter)*: 1000

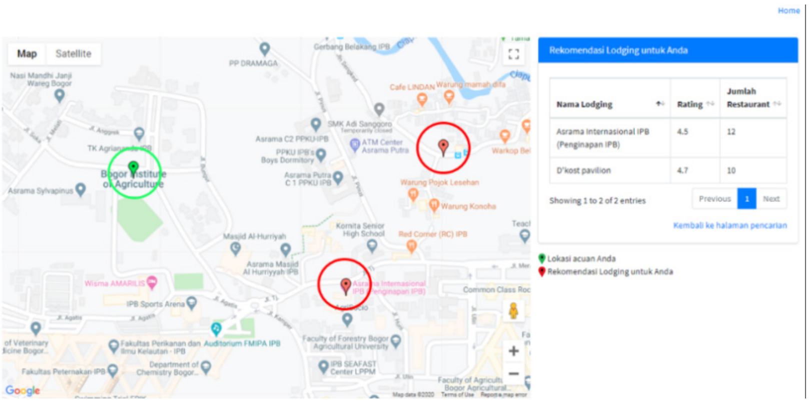
Jenis Fasilitas Sekitar Objek*: × Restaurant

Maksimum Radius Fasilitas Sekitar (meter)*: 500

Minimum Rating Fasilitas Sekitar*: Tidak ada

*harus diisi

Figure 4. The web interface of form page



Rekomendasi Lodging untuk Anda

Nama Lodging	Rating	Jumlah Restaurant
Asrama Internasional IPB (Penginapan IPB)	4.5	12
D'kost pavilion	4.7	10

Showing 1 to 2 of 2 entries

● Lokasi acuan Anda
● Rekomendasi Lodging untuk Anda

Figure 5. The web interface of result page

Figure 4 shows IPB University with green markers as the reference point to select interesting rent house with a maximum radius of 1 km. Facilities to be considered around rent house are restaurants with a maximum radius of 0.5 km without considering restaurant ratings. After processing, in Figure 5 shows the skyline rent house with red markers, IPB International Dormitory and D'kost Pavilion. Results form in Figure 5 also contain information about the number of restaurant around recommended skyline rent house along with the rating.

e. System Testing

This stage is carried out by testing the system that have been built using Blackbox testing method. Blackbox testing method is a software testing method that focuses on system functionality, specifically on the input and output without testing the algorithm. The test conducted was a system interaction from filling the form to producing a list of location recommendations based on input on the form.

f. Experiments

System performance tests are related to location search time on Google Maps. We conducted some experiments with these scenarios:

- Scenario 1: to test system performance towards the increase of number of surrounding facilities. We set the number of candidate objects to 5 objects, and increase the surrounding facilities by 2, 5, 10, 15, and 20.
- Scenario 2: to test system performance towards the increase of the radius (distance) of surrounding facilities from reference point Q. We fixed the number of candidate objects and reduce or enlarge the radius from reference point.
- Scenario 3: to test system performance towards the increase of number of surrounding facility type. Since the number of facility types is the number of dimension of candidate objects, we varies the number of facility type to 2D, 4D, 6D and 8D and varying the size of data for each facility from 5 to 20 data.
- Scenario 4: to test the effect of data collection module to the entire system performance.

3. Result and Discussion

a. System Testing

Blackbox testing method is used to test the results of the web that has been developed. Based on all the test results, it can be seen that all functions in the location search feature based on the type and number surrounding facilities have been successfully implemented. The implementation have included the form filling function to get a list of location recommendations based on input on the form. The results of the Blackbox test can be seen in Table 3 and code has been published in Github repository at https://github.com/salsakhairinaa/BasedSurroundingSkyline.git.

Table 3. Example of data collection module results

Testing Name	Testing Conditions	Test Result
Select the location search feature	If the user clicks on the location search feature based on the type and number of facilities around.	The Location Search form page appears.
Input the data	If all fields on the form have been filled in, then the user clicks the "Process" button.	The system will display a result page that lists the location recommendations along with rating information and the number of facilities to the user.
	If all fields on the form are not filled in, then the user clicks the "Process" button.	The system will remain on the form page and the message "Please complete the data" appears.

g. Experiments

Tests are performed to see the system performance measured in time required to complete the search. In the first scenario, the number of candidate objects is fixed at 5 objects, while the type of surrounding facilities is varied from 2, 5, 10, 15, to 20. The second scenario is to observe the effect of different radius on the system performance by setting two radius distance: a small radius (500 meters), and a large radius (5000 meters), while the number of candidate objects are fixed.

Each experiment was carried out with 10 iterations, and the result of each iteration is recorded. Table 4 shows the results of scenario 1 and scenario 2 tests. Figure 6 shows the results of the average execution time in seconds (d) which reveals that the execution time increases along with the increase in the number of facility type. The experiment also reveals that there is no big difference in the average execution time for small and large radius. This shows that the time required for location search based on surrounding facilities is greatly influenced by the number of facility types that are taken into account.

Figure 7 displays the experiment results from scenario 3. Figure 7 shows that the running time of the SFS algorithm increases with increasing data size. The execution time also increases with increasing data dimensions or the number of facility types around the candidate object. It can be seen that adding the data collection module in the system significantly increases processing time of location search using SFS algorithm.

Figure 8 shows experiment results of scenario 4, the execution time of the SFS algorithm which considers number of facilities. The data used in the SFS algorithm is set to 20 candidate objects with the number of facilities considered are 2, 4, 6, and 8. Figure 8 (a) is the execution time of the SFS algorithm in finding the location from existing data. Figure 8 (b) is the execution time starting from collecting data using Google Maps until determining the location using SFS. Based on the execution time, it can be concluded that the execution time increases with the increasing number of facilities being considered. The data used in this study are dynamic data so that the execution time in this study increases because of the data collection process, but this does not affect the performance of the SFS algorithm. The complexity of the SFS algorithm in the best case state is $O(dn + n\log n)$ and in the worst case is $O(dn^2)$, where d is the number of dimensions or number of facilities considered and n data size or number of candidate objects.

Table 4. Average of execution time and radius

Iterations	Number of Attributes					Radius	
	2	5	10	15	20	500	5000
1	11.54	48.46	96.64	116.47	203.33	14.42	14.51
2	10.06	45.18	95.38	184.06	157.59	13.60	13.33
3	9.40	45.39	83.63	125.23	205.47	14.27	14.73
4	10.70	55.01	97.71	103.20	204.02	14.12	15.49
5	10.11	57.44	98.26	105.44	199.63	13.32	14.84
6	13.77	53.48	93.99	115.34	201.49	14.44	13.27
7	13.74	55.49	94.90	113.29	197.62	12.02	16.47
8	9.49	44.51	88.48	110.68	201.98	12.72	15.87
9	10.91	56.27	88.73	118.45	199.93	14.31	13.51
10	9.02	56.08	87.98	111.14	201.09	13.75	14.02
Average of execution time (s)	10.87	51.37	92.57	120.33	197.21	13.70	14.60

4. Conclusion

This research has succeeded in implementing the SFS algorithm in Google Maps to answer location selection based on surrounding facilities query. The results of the study are a web-based application with simple user interface so that Google Maps users can run the query easily. The time required to search for a location depends on the number of facility types considered by the user. SFS algorithm performance is affected by the increase in data size and data dimensions.

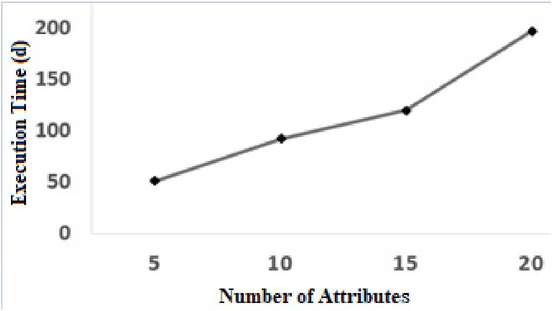


Figure 6. The average execution time based on increasing number of attributes

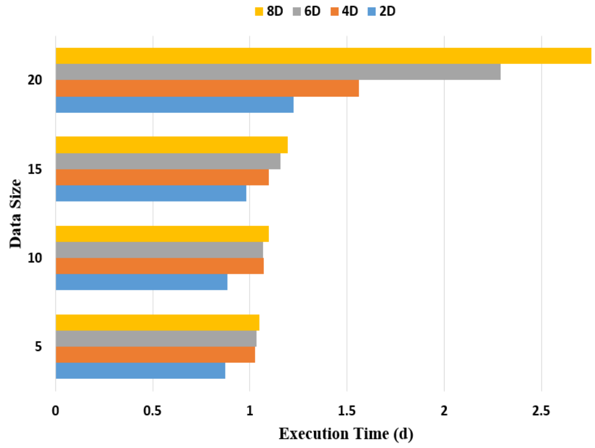
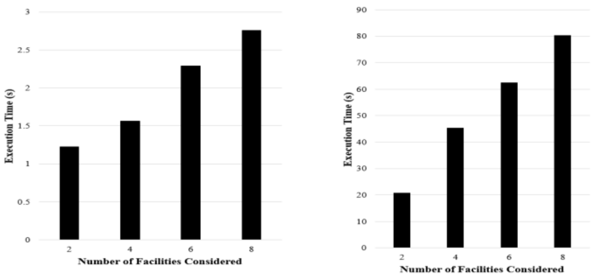


Figure 7. The average SFS execution time based on number of data and facilities



(a) **(b)**

Figure 8. (a) SFS algorithm execution time (b) The execution time of the data collection module is continued by the SFS module

Reference

[1] Z. Chang, M.S. Arefin, Y. Morimoto, Hotel recommendation based on surrounding environments, In Second IIAI International Conference on Advanced Applied Informatics, 2013, pp. 330-336.

[2] A. Syafrianto, "A Development of Spatial Skyline Query Based on Surrounding Environment for Data Streaming Using Apache-Spark", M.Kom. thesis, Computer Science, IPB University, Bogor, ID, 2010.

[3] G. Popovic, D. Stanujkic, M. Brzakovic, and D. Karabasevic, A multiple-criteria decision-making model for the selection of a hotel location. Land use policy, 2019, pp.49-58.

[4] S. Borzonyi, D. Kossmann, and K. Stocker, The skyline operator, In Proc. of ICDE, 2001, pp. 421-430.

[5] K.L. Tan, P.K. Eng, and B.C. Ooi, Efficient progressive skyline computation In Proc. of VLDB Conference, 2001, pp. 301-310.

[6] D. Kossmann, F. Ramsak, and S. Rost, Shooting stars in the sky: An online algorithm for skyline queries, In Proc. of VLDB Conference, 2002, pp. 275-286.

[7] D. Papadias, Y. Tao, G. Fu, and B. Seeger, An optimal and progressive algorithm for skyline queries, In Proc. of ACM SIGMOD Conference, 2003, pp. 467-478.

[8] J. Chomicki, P. Godfrey, J. Gryz, and D. Liang, Skyline with Presorting: Theory and Optimizations, In Proc. of the international IIS: IIPWM'06 conference, 2006, pp. 595-604.

[9] S. Shah S, A. Thakkar, S. Rami, A Survey paper on skyline query using recommendation system, In Journal of Data Mining & Emerging Technologies, 2016, pp. 1-6.

[10] M. Sharifzadeh, and C. Shahabi, The spatial skyline queries, In Proc. of VLDB, 2006, pp. 751-762.

[11] W. Son, M. Lee, H. Ahn, and S. Hwang, Spatial skyline queries: an efficient geometric algorithm, In Proc. of SSTD,2009, pp. 247-264.

[12] X. Guo, Y. Ishikawa, and Y. Gao, Direction-based spatial skylines, In Proc. of ACM SIGMOD Conference, 2010, pp. 73-80.

[13] K. Deng, X. Zhou, and H.T. Shen, Multi-source skyline query processing in road networks In Proc. of ICDE, 2007, pp. 796-805.

[14] M. Safar, D.E. Amin, and D. Taniar, Optimized

skyline queries on road networks using nearest neighbors, In Journal of Personal and Ubiquitous Computing, vol. 15, issue 8, 2011, pp. 845-856.

[15] Y.K. Huang, C.H. Chang, and C. Lee, Continuous distance-based skyline queries in road networks, In Journal of Information Systems, vol. 37, 2006. pp. 611-633.

[16] M.S. Arefin, Jinhao X, Zhiming C, Morimoto Y, Skyline query for selecting spatial objects by utilizing surrounding objects, In Journal of Computers, 2013, pp. 1742-1747.

[17] T. Djatna, F.H. Putra, dan A. Annisa, An Implementation of Area Skyline Query to Select Facilities Location Based on User's Preferred Surrounding Facilities. In Proc. of IEEE conference, ICACSIS, 2020, pp. 15-20.

[18] C. Li, A. Annisa, A. Zaman, M. Qaosar, S. Ahmed, and Y. Morimoto, Mapreduce algorithm for location recommendation by using area skyline query. In Algorithms, 11(12), 2018, pp.191.

[19] L.G. Asri, and A. Annisa, Application of Skyline Query on Route Selection (the Case Study of Bogor City Roadway). In the Proc. of IEEE conferences, International Conference on Computer Science and Its Application in Agriculture (ICOSICA), 2020, pp. 1-6.

[20] A. Annisa, A. Zaman, and Y. Morimoto, Area skyline query for selecting good locations in a map. Journal of Information Processing, 24(6), 2016, pp.946-955.

[21] D. Papadias, P. Kalnis, J. Zhang, and Y. Tao, Efficient OLAP operations in spatial data warehouses, In Lecture Notes in Computer Science, 2001, vol. 2121, pp. 443-459.

[22] J. Chomicki, P. Godfrey, J. Gryz, and D. Liang, Skyline with presorting, In Proc. of ICDE, 2003, pp. 717-816.

[23] E. Costa-Montenegro, F. J. González-Castaño, D. Conde-Lagoa, A. B. Barragáns-Martínez, P. S. Rodríguez-Hernández and F. Gil-Castiñeira, QR-Maps: An efficient tool for indoor user location based on QR-Codes and Google maps, In 2011 IEEE Consumer Communications and Networking Conference (CCNC), 2011, pp. 928-932.

[24] P. Pokorný, P., 2017, Determining Traffic Levels in Cities Using Google Maps. In Proc. of IEEE, The Fourth International Conference on Mathematics and Computers in Sciences and in Industry (MCSI), 2017, pp. 144-147.

[25] M.H. Erol and F. Bulut, Real-time application of travelling salesman problem using Google Maps API. In Proc. of IEEE, Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT), 2017, pp. 1-5.

[26] C. Costa, J. Ha, and S. Lee, Spatial disparity of income-weighted accessibility in Brazilian Cities: Application of a Google Maps API. Journal of Transport Geography, *90*, 2021, p.102905.

[27] T. Listyorini and S. Muzid, Population resizing on fitness improvement genetic algorithm to optimize promotion visit route based on android and google maps API. In AIP Conference Proceedings, Vol. 1855, No. 1, 2017, p. 060001.

[28] K. Kodama, Y. Iijima, X. Guo, and Y. Ishikawa, Skyline queries based on user locations and preferences for making location-based recommendations, In Proc. of ACM LBSN, 2009, pp. 9-16.

[29] R.C. Wong, A.W. Fu, J. Pei, Y.S. Ho, T. Wong, and Y. Liu, Efficient skyline querying with variable user preferences on nominal attributes, In Proc. of VLDB, 2008, pp. 1032-1043.

[30] C. Kalyvas and T. Tzouramanis, A survey of skyline query processing, In arXiv preprint arXiv:1704.01788, 2017, pp. 19-20.

khazanah informatika

Jurnal Ilmu Komputer dan Informatika

DenseNet-CNN Architectural Model for Detection of Abnormality of Acute Pulmonary Edema

Cynthia Hayat
Correspondence: cynthia.hayat@ukrida.ac.id
Department of Information System
Krida Wacana Christian University
West Jakarta, Indonesia

Abstract-Acute pulmonary edema (EPA) is a condition of emergency respiratory distress that results from the sudden and rapid build-up of fluid into the lungs. Rapid screening of EPA patients is necessary so that radiologists can make the prognosis as early as possible. In addition, reliance on the expert's knowledge of reasoning also hinders the diagnostic process. This research was conducted by developing an architectural model for machine learning systems with a deep learning approach. With the concept of representative learning, the denseNet-CNN algorithm connects each layer to another by means of a feed-forward. The data used is Image CXR-14 specifically labeled pulmonary edema pathology. The size of each CXR-14 image is 1024 × 1024 with a value of 8 bits grayscale. The size of each CXR-14 image is 1024 × 1024 with a value of 8 bits grayscale. The architectural model development stages consist of the preparation stage, data resampling, data training and data testing. Optimizer parameters used are Adam's optimizer, learning rate of 0.0001 and weight decay = 1e-5 and the loss used is binary cross entropy. The resulting mean AUROC analysis showed the sensitivity value of the 10% dataset was 71.493% and the specificity value of 10.011% was obtained at the second hold of the k-fold cross validation method after holdout validation, so that the resulting model was valid. The detection system developed from the denseNet-CNN model is expected to help radiologists identify abnormalities in CXR images quickly, precisely, and consistently. The denseNet-CNN model is also developed in the form of a heatmap visualization by localizing the features you are looking for. With localization in the form of a heat map, detection of pathological abnormalities of PEA is easier to do and to be recognized.

Keywords: denseNet-CNN, EPA, abnormality detection

Article info: *submitted July 20, 2020, revised September 6, 2020, accepted November 2, 2020*

1. Introduction

Acute pulmonary edema (EPA) is a condition of urgency characterized by a rapid respiratory emergency caused by displacement and accumulation of fluid in the lungs. According to the source of the cause, EPA is divided into two types, namely cardiogenic EPA and noncardiogenic EPA. The high prevalence rate of EPA can be seen from a study involving approximately 600 hospitals in Europe, Latin America, and Australia. Data show EPA is present in 37% of patients with acute heart failure [1][2].

Fast and precise screening for patients with EPA cases is needed so that doctors can make the prognosis as early as possible. One of the initial screenings performed on EPA patients is by doing a chest X-ray examination, known as ChestX-Ray (CXR) screening. Generally, the reading and diagnosis of CXR images is performed by a radiologist by comparing the CXR images of EPA patients with normal CXR images to detect abnormalities [3 [4].

Dependence on the knowledge of an expert radiologist regarding the principles of anatomy, physiology, and pathology is a factor that can hinder making a diagnosis as early as possible [5]. Another difficulty in the process of detecting CXR images is the difficulty of expert radiologists in developing consistent reasoning techniques in reading CXR images while considering all common chest diseases that require a long time to diagnose a CXR image [6].

One solution to solve this problem is to develop a machine learning system. Amit Kumar Jaiswal's research, entitled Identifying Pneumonia in Chest X-Rays: A Deep Learning Approach uses the Mask-RCNN method, used a deep neural network that combines global and local features for pixel-based segmentation. The identification model proposed in the study achieved reasonably good performance after evaluating a dataset of chest radiographs depicting potential causes of pneumonia [7].

The approach recommended in this study is to use deep learning (DL) convolutional neural network models. The DL approach adopts the way the human brain works in managing data as representative learning which is then classified into layers. Research by Huang, Liu, Weinberger, & van der Maaten conducted in 2017 shows the importance of layer depth, better accuracy, and efficient training will be achieved if CNN has a closer connection between the layers that are close to the input and the layers that are close to it. with output [8]. DenseNet has an architecture that connects each layer to another in a feed-forward manner. Densenet itself has several advantages, namely: reducing the vanishing-gradient problem, strengthening feature propagation, reusing features, and reducing the number of parameters [9][10][11].

The computer-assisted detection system is expected to help radiologists identify abnormalities in CXR images quickly, precisely, and consistently. Computers can be taught to read and process a very large number of CXR image scans in a short amount of time. to confirm the results found by the radiologist and potentially identify other findings that may have been found. The resulting model is an artificial intelligence mechanism that can direct radiologists to make better diagnostic decisions for patients.

2. Methods

a. Data collection

The CXR images used in the CNN architectural model developed in the study were obtained from the public dataset of NIH Clinical Center, a clinical research hospital for the National Institutes of Health based in Maryland - United States, called ChestX-ray14 (CXR-14). This public dataset is the largest CXR dataset available to the public, containing 112,120 anonymous CXR front view images derived from 30,805 patients including patients with advanced lung disease. Each dataset summary page also contains the license terms and citation requirements which can be accessed on TCIA datasets from Cloud Storage, BigQuery, or by using the NIH Clinical Center's Cloud Healthcare API [12][13]. One of the CXR-14 images used is shown in Figure 1.

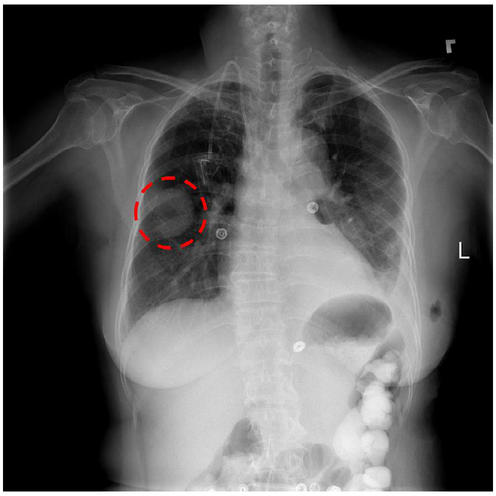


Figure 1. CXR image that identifies lung masses

The size of each CXR-14 image is 1024 × 1024 with a value of 8 bits grayscale as shown in table 1.

Table 1. Number of positive labels per pathology in the CXR-14 dataset

No.	Pathology	Number of Positive Labels	Percentage of the amount
1	Edema	2,303	2.05%

The number of positive labels for pulmonary edema was 2,303 with the assumption that there would be redundancy of patient data between existing pathology labels.

b. DenseNet CNN Procedure

The proposed denseNet-CNN model development procedure is as follows:

Step 1: The preparation stage

In this preparation stage, analysis is carried out first to adjust the Deep Learning Framework to be applied with the availability of hardware in conducting training, validation and testing of the CNN model to be developed, as for the specifications of the hardware to be used are as follows:

- Desktop CPU: Intel (R) Core (TM) i3-8100 CPU @ 3.60GHz
- Memory: 8052MB
- VGA: NVIDIA Corporation GP107 [GeForce GTX 1050] Storage Media: 256GB Solid State Disk

Step 2: Re-Sampling Data

Modifications to the algorithm applied to the CXR-14 PEA dataset resulted in an underfit model due to imbalance data, therefore re-sampling the CXR-14 dataset, in order to provide valid results. The process of re-sampling this dataset begins with training on less data, which in this study was determined as much as 10% of the total dataset, which includes: all data with a positive PE label found on CXR-14, and the rest randomly chooses data labeled PE negative. [14] Re-sampling the dataset obtained is separated systematically using Python coding into training data, validation data and testing data. Resampling Dataset for Hold-out validation is illustrated in Figur 2 & 3.

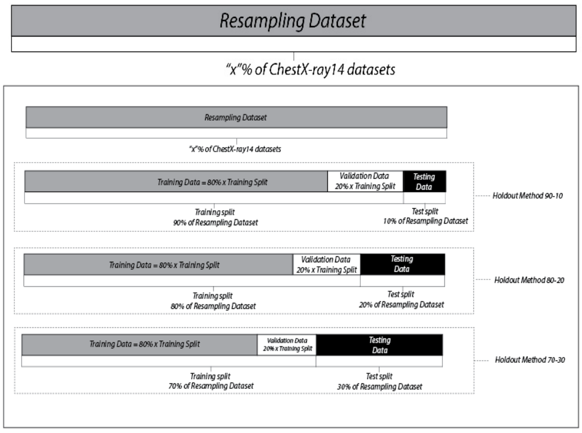
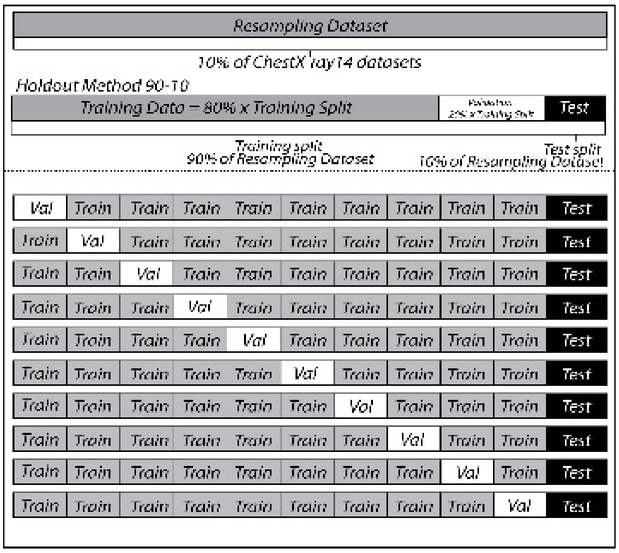

Resampling Dataset
"x"% of ChestX-ray14 datasets
Resampling Dataset
"x"% of ChestX-ray14 datasets
Training Data = 80% x Training Split
Validation Data 20% x Training Split
Testing Data
Training split 90% of Resampling Dataset
Test split 10% of Resampling Dataset
Holdout Method 90-10
Training Data = 80% x Training Split
Validation Data 20% x Training Split
Testing Data
Training split 80% of Resampling Dataset
Test split 20% of Resampling Dataset
Holdout Method 80-20
Training Data = 80% x Training Split
Validation Data 20% x Training Split
Testing Data
Training split 70% of Resampling Dataset
Test split 30% of Resampling Dataset
Holdout Method 70-30

Figure 2. Resampling Dataset for Hold-out validation


Resampling Dataset
10% of ChestX ray14 datasets
Holdout Method 90-10
Training Data = 80% x Training Split
Validation 20% x Training Split
Test
Training split
90% of Resampling Dataset
Test split
10% of Resampling Dataset

Val	Train	Train	Train	Train	Train	Train	Train	Train	Train	Test
Train	Val	Train	Train	Train	Train	Train	Train	Train	Train	Test
Train	Train	Val	Train	Train	Train	Train	Train	Train	Train	Test
Train	Train	Train	Val	Train	Train	Train	Train	Train	Train	Test
Train	Train	Train	Train	Val	Train	Train	Train	Train	Train	Test
Train	Train	Train	Train	Train	Val	Train	Train	Train	Train	Test
Train	Train	Train	Train	Train	Train	Val	Train	Train	Train	Test
Train	Train	Train	Train	Train	Train	Train	Val	Train	Train	Test
Train	Train	Train	Train	Train	Train	Train	Train	Val	Train	Test
Train	Train	Train	Train	Train	Train	Train	Train	Train	Val	Test

Figure 3. Resampling Dataset for Hold-out validation

Step 3: Training Stage

Furthermore, experiments and validations were carried out on the CheXNet scripting sources found on the github repository portal, this study determines the implementation of the PyTorch DL Framework made by Andrey G. (zoogzog) on the link https://github.com/zoogzog/chexnet . Referring to the selected scripting, some environmental elements for the DL framework need to be prepared first, namely using: [12]

- Operating System: Ubuntu 17.10
- Compilers: C (GCC) 7.2.0
- CUDA compilation tools: NVIDIA CUDA, V8.0.61

Experiments were carried out by replicating Python coding in the PyTorch environment that implements training and validation functions so that the model is in the form of a file that will be used in the testing stage. The optimizer used is Adam's algorithm as a standard parameter that can produce output quickly. The learning rate used is 0.0001, the weight decay is 1e-5, and the loss used is the cross entropy loss.

Step 4: Testing Phase

Furthermore, the testing process is carried out to produce AUROC performance against the PEA pathology contained in CXR-14. The distribution of the amount of training, validation, and testing data used in scripping for the DL framework is shown in table 2.

Table 2. Distribution of Total Data Training, Validation and Testing on Scripting

Image CXR	Training process	Validation process	Testing process
amount	78,468 data	11,219 data	22,433 data

Step 5: The resulting output

The AUROC performance obtained at this preparation stage is then compared with the results obtained by Rajpurkar et.al., in their journals as a reference in proceeding to the research implementation stage, as shown in table 3[15].

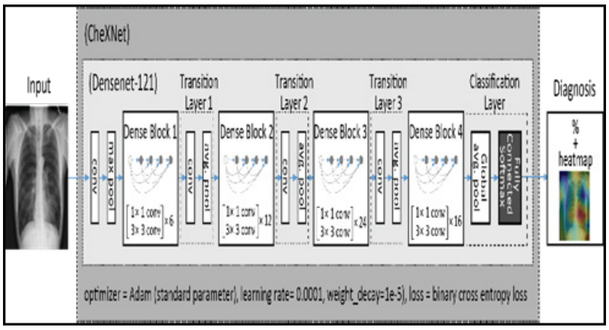
Table 3. Comparison of the results of testing the DenseNet algorithm between the results of Rajpurkar et.al., and the results of testing scripts for pulmonary edema

Patology	CheXNet-14	Result
Acute Pulmonary Edema	0.8878	0.9017
AUROC mean	0.841	0.8508

CNN's DenseNet Architecture Model Modifications

The architectural model developed using the DenseNet CNN algorithm which has advantages in implementing the depth of a layer, better accuracy, and efficient training will be achieved if CNN has a closer connection between the layer close to the input and the layer close to the output. Densenet CNN has an architecture that connects each layer to another by means of feed-forward [16][17][18].

The modification of CNN's DenseNet architectural model in this study can be seen in Figure 4.



optimizer = Adam (standard parameter), learning rate= 0.0001, weight_decay=1e-5), loss = binary cross entropy loss

Figure 4. CNN's DenseNet Architectural Model for EPA Disease Abnormality Detection

Generally, the CNN architecture consists of the input layer, hidden layer and output layer stages. CNN's DenseNet architecture above by skipping processes in hidden layers. The effect of the skip hidden layer is that the loss value from the previous layer can be carried to the next network so that the loss after the skip process is a combination of loss with the loss brought before by the skip connection. In this study, the skip connection added to the CNN architecture with the DenseNet algorithm can minimize the vanishing-gradient problem so that feature propagation can be strengthened, repetitive features on features, and reduces the use of parameters in the architectural model being developed.

Adam Optimizer was chosen because it is the most popular algorithm and it can produce output faster and better than other methods. The optimization done by Adam can be used as a substitute for the classic stochastic gradient descent procedure so that it can update the network weight repetitively on features based on training data. The stochastic gradient descent function maintains one learning rate for all weight updates used, where the learning rate will not change during the training process. [19] [20]

The learning rate used is 0.0001. The learning rate is maintained for each network weight (parameter) and is adapted separately as learning unfolds. With Adam's algorithm it is also possible to calculate individual adaptive learning rates for parameters different from the predicted first and second moments of the gradient. [21]

In the classification layer, a full connected softmax and a global average pool will be generated where the diagnoses of how many abnormalities are detected in the CXR-14 heatmap image.

2. Result

a. Holdout Validation at 10% Chest X-ray 14 Dataset

The distribution of the dataset for this method is carried out in accordance with the value of the distribution of training, validation and testing data shown in table 4, with the method scheme illustrated in Figure 5 with a value of x = 10.

Table 4. Holdout Validation Method Dataset Resampling Label on 10% Dataset CXR-14

Method	Data Training	Data Validation	Data Testing
Holdout 90-10	8.072	2.018	1.121
Holdout 80-20	7.175	1.793	2.242
Holdout 70-30	6.278	1.569	3.363

The training is carried out using the same Python programming as the 100% dataset training. The best results of modeling the three holdout validation methods at 10% of the ChestX-ray14 dataset are shown in table 4. The best mean AUROC value achieved is 0.9114 with Binary Cross Entropy (BCELoss) between the target and the output is 0.3238 which is obtained from the 90- division. 10. , determining the amount of training data, validation and schematic testing is shown in Figure 5. Graphic images of BCELoss on each batch of epoch validation are shown in Figure 6.

The results of model accuracy with three hold-out validation methods at 10% of the CXR-14 dataset are shown in table 5.

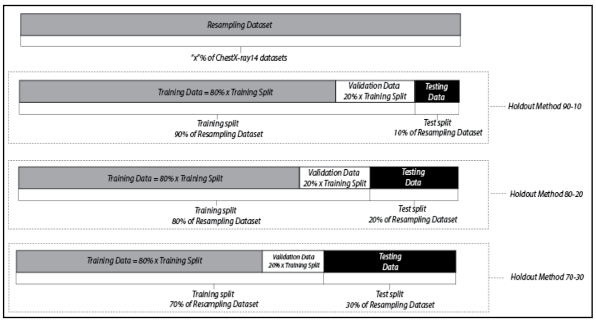
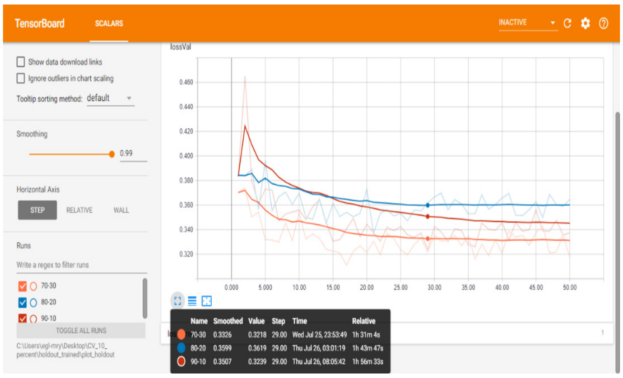

Resampling Dataset
"x"% of ChestX-ray14 datasets
Training Data = 80% x Training Split
Validation Data 20% x Training Split
Testing Data
Holdout Method 90-10
Training split 90% of Resampling Dataset
Test split 10% of Resampling Dataset
Training Data = 80% x Training Split
Validation Data 20% x Training Split
Testing Data
Holdout Method 80-20
Training split 80% of Resampling Dataset
Test split 20% of Resampling Dataset
Training Data = 80% x Training Split
Validation Data 20% x Training Split
Testing Data
Holdout Method 70-30
Training split 70% of Resampling Dataset
Test split 30% of Resampling Dataset

Figure 5. Schematic of determining the training data, validation and testing of the hold-out validation method



□ Show data download links

□ Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing 0.99

Horizontal Axis: STEP

Runs

Write a regex to filter runs

☑ ○ 70-30

☑ ○ 80-20

☑ ○ 90-10

C:\Users\egl-mry\Desktop\CV_10_percent\holdout_trained\plot_holdout

	Name	Smoothed	Value	Step	Time	Relative
●	70-30	0.3326	0.3218	29.00	Wed Jul 25, 23:53:49	1h 31m 4s
●	80-20	0.3599	0.3619	29.00	Thu Jul 26, 03:01:19	1h 43m 47s
●	90-10	0.3507	0.3239	29.00	Thu Jul 26, 08:05:42	1h 56m 33s

Figure 6. Graph BCELoss holdout validation method on 10% of ChestX-ray14 dataset

Table 5. Holdout Validation Model Accuracy Results on 10% Chestx-Ray 14 Dataset

Method	AUROC mean	BCELoss	Epoch#
Holdout 90-10	**0.9114**	**0.3238**	**29**
Holdout 80-20	0.8992	0.3369	21
Holdout 70-30	0.8940	0.3110	17

The number of positive PE data labels and negative PE data labels on the 90-10 holdout validation method at 10% of the ChestX-ray14 dataset is mentioned in table 6.

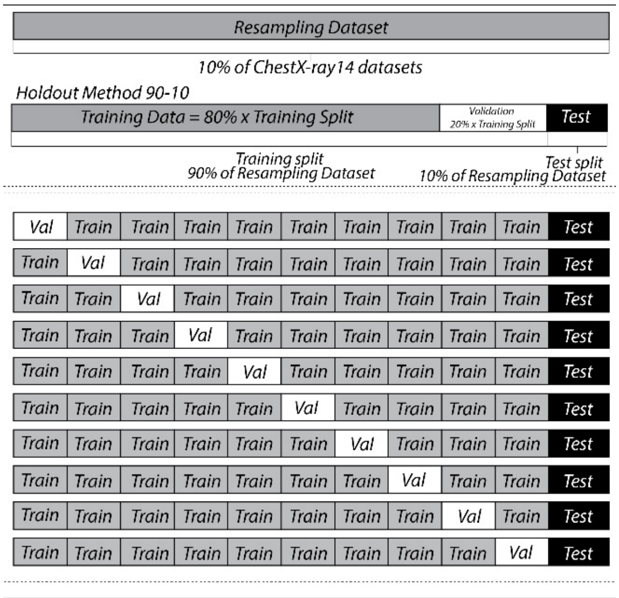
Table 6. Number of PE detection labels on Modified 90-10 Holdout Validation Dataset on 10% Chestx-Ray 14 Dataset

	Training	Validation	Testing	Total
Label Positif	1.692	389	222	2.303
Label Negatif	6.380	6.380	1.629	899
Total	8.072	8.072	2.018	1.121

b. Application of the K-fold cross validation over holdout validation method on 10% of the ChestX-ray 14

Furthermore, the K-fold cross validation over holdout validation method is applied, with a value of k = 10 (10 fold) based on the best 10% ChestX-ray14 dataset that has been obtained from the holdout validation method previously carried out, namely in the 90-10 division, determining the amount of data. training, validation and schematic testing are shown in Figure 7.

The results obtained from the modeling method of 10 fold cross validation over holdout validation on 10% of the ChestX-ray14 dataset are shown in Table 7. The best mean AUROC produced was 0.9164, with a BCELoss value of 0.3167, which was achieved by the 2nd fold in the 42nd batch epoch validation of the 50 defined epoch batches. Graphic of BCELoss achievement in each epoch batch is shown in Figure 8.


Resampling Dataset
10% of ChestX-ray14 datasets
Holdout Method 90-10
Training Data = 80% x Training Split
Validation 20% x Training Split
Test
Training split 90% of Resampling Dataset
Test split 10% of Resampling Dataset

Val	Train	Train	Train	Train	Train	Train	Train	Train	Train	Test
Train	Val	Train	Train	Train	Train	Train	Train	Train	Train	Test
Train	Train	Val	Train	Train	Train	Train	Train	Train	Train	Test
Train	Train	Train	Val	Train	Train	Train	Train	Train	Train	Test
Train	Train	Train	Train	Val	Train	Train	Train	Train	Train	Test
Train	Train	Train	Train	Train	Val	Train	Train	Train	Train	Test
Train	Train	Train	Train	Train	Train	Val	Train	Train	Train	Test
Train	Train	Train	Train	Train	Train	Train	Val	Train	Train	Test
Train	Train	Train	Train	Train	Train	Train	Train	Val	Train	Test
Train	Train	Train	Train	Train	Train	Train	Train	Train	Val	Test

Figure 7. Schematic of determining training data, validation and testing of the k-fold cross validation method

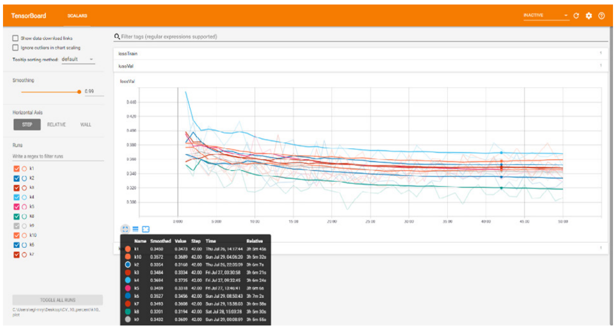


Figure 8. Graph of BCELoss k-fold validation method on 10% of the ChestX-ray 14 dataset

Table 7. Modeling Accuracy of the K-Fold Cross Validation Method Over Holdout Validation on the CXR-14 10% Dataset

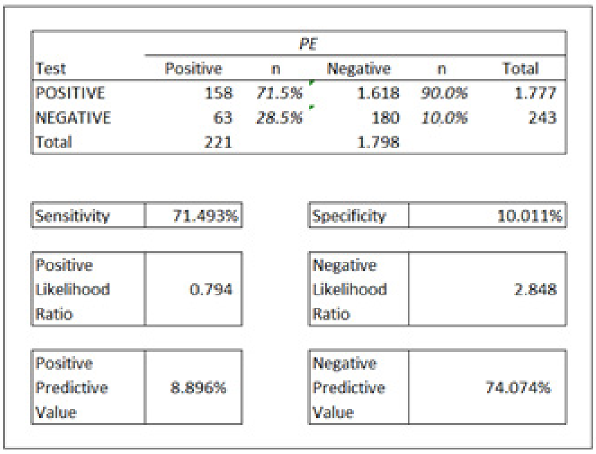
Method	AUROC mean	BCELoss	Epoch#
Fold-1	0.9082	0.3225	27
Fold-2	**0.9164**	**0.3167**	**42**
Fold-3	0.9154	0.3274	24
Fold-4	0.9079	0.3484	19
Fold-5	0.9082	0.3226	18
Fold-6	0.9056	0.3249	25
Fold-7	0.9100	0.3291	17
Fold-8	0.9125	0.2894	22
Fold-9	0.9157	0.3190	25
Fold-10	0.9118	0.3319	26

c. AUROC analysis of 10% ChestX-ray 14 dataset

The area under receiver operating characteristics (AUROC) is a performance metric that you can use to evaluate a classification model. The analysis was to determine the sensitivity of the best model produced, namely the second fold of the k-fold cross validation method after holdout validation. The calculation of the mean AUROC is shown in Figure 9 and a slice of the output dataset is shown in Table 8.

Table 8. AUROC calculation for a positive label and a negative label on the CXR-14 image

Line #	AUROC	Line #	AUROC
1	0.9152	1	0.0244
2	0.8589	2	0.6133
3	0.8710	3	0.0576
…	…	…	…
220	0.8219	1796	0.0054
221	0.2529	1797	0.0417
222	0.7647	1798	0.1170
Pieces of AUROC calculation data for detection of positive PE labels in 10% of the CXR-14 dataset		Pieces of AUROC calculation data for detection of positive PE labels in 10% of the CXR-14 dataset	



	PE				
Test	Positive	n	Negative	n	Total
POSITIVE	158	71.5%	1.618	90.0%	1.777
NEGATIVE	63	28.5%	180	10.0%	243
Total	221		1.798		

Sensitivity	71.493%	Specificity	10.011%
Positive Likelihood Ratio	0.794	Negative Likelihood Ratio	2.848
Positive Predictive Value	8.896%	Negative Predictive Value	74.074%

Figure 9. The results of AUROC calculations on 10% of the ChestX-ray 14 dataset

From the calculation results, it is known that the sensitivity value has increased significantly so that it is above the cut-off value, with compensation there is an increase in the specificity value, which means that the resulting model has increased the likelihood of getting a true positive value when detecting a CXR-14 image with a PE label. positive is better, although there is a decrease in the confidence value of the negative PEA label training results.

With the results of the analysis obtained, it is concluded that the resulting model is valid, with a better sensitivity value obtained from the 10% dataset of 71.493% and a specificity value of 10.011%. The comparison of the model test results is shown in table 9.

Table 9. Model Measurement Results

	10% dataset on (k-fold 2)
Total Dataset	11.211
Data Training	8.072
Data Validation	2.018
Data Testing	1.121
AUROC mean	0.9164
Sensitivity	71.493%
Specificity	10.011%

The research continues to the next stage by bringing the best model file obtained, from modeling 80-20 holdout validation on 10% of the CXR-14 dataset as a model for testing CXR input in detecting PEA pathology by displaying a heat map to visualize areas on CXR-14, where disease is present, by using the Class Activation Mapping. Implementing this application can assist medical personnel in localizing the features they looking for.

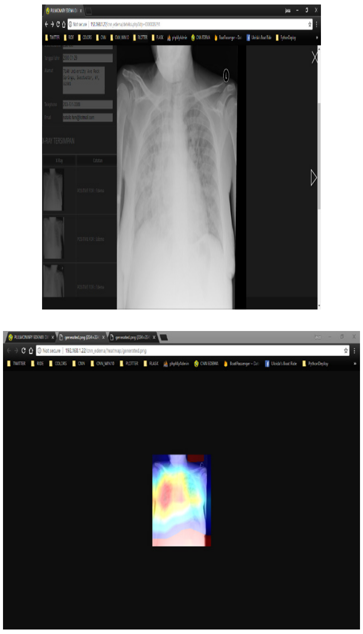


Figure 10. The result of heat map visualization on the resulting CNN Dense-Net model

3. Conclusion

The valid model was obtained in the 6th modification of the second k-fold with a mean AUROC value of 0.9164, a better sensitivity value was obtained from the 10% dataset of 71.493% and a specificity value of 10.011%. Optimizer parameters used are Adam's optimizer, learning rate of 0.0001 and weight decay = 1e-5 and the loss used is binary cross entropy.

The denseNet-CNN model is also developed in the form of a heatmap visualization by localizing the features to be searched. The architecture is able to recognize predictive information with localization in the form of a heat map. This makes easier to detect and recognize PEA pathologic abnormalities.

Reference

[1] S. H. Rampengan, "EDEMA PARU KARDIOGENIK AKUT," J. BIOMEDIK, 2014, doi: 10.35790/jbm.6.3.2014.6320.

[2] H. Nendrastuti and M. Soetomo, "Edema Paru Akut Kardiogenik Dan Non Kardiogenik," Maj. Kedokt. Respirasi, 2010.

[3] M. Irawaty, "Penatalaksanaan Edema Paru pada Kasus VSD dan Sepsis VAP Treatment of Lung Oedema in VSD and VAP Sepsis," Anest. Crit. Care, 2010.

[4] A. Jatu and Lusiana, "Peranan Epitel Alveoli pada Edema Paru Non-kardiogenik," Ckd, 2015.

[5] C. Hayat and B. Abian, "The modeling of artificial neural network of early diagnosis for malnutrition with backpropagation method," 2018, doi: 10.1109/IAC.2018.8780505.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2016, doi: 10.1109/CVPR.2016.90.

[7] A. K. Jaiswal, P. Tiwari, S. Kumar, D. Gupta, A. Khanna, and J. J. P. C. Rodrigues, "Identifying pneumonia in chest X-rays: A deep learning approach," Meas. J. Int. Meas. Confed., 2019, doi: 10.1016/j.measurement.2019.05.076.

[8] G. Huang, S. Liu, L. Van Der Maaten, and K. Q. Weinberger, "CondenseNet: An Efficient DenseNet Using Learned Group Convolutions," 2018, doi: 10.1109/CVPR.2018.00291.

[9] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2017, doi: 10.1109/CVPR.2017.243.

[10] T. NURHIKMAT, "IMPLEMENTASI DEEP LEARNING UNTUK IMAGE CLASSIFICATION MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN) PADA CITRA WAYANG GOLEK," UNIVERSITAS ISLAM INDONESIA. 2018.

[11] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," 2018, doi: 10.1109/ICEngTechnol.2017.8308186.

[12] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "ChestX-ray8: Hospital-

scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," 2017, doi: 10.1109/CVPR.2017.369.

[13] E. T. Nader, "Chest X-ray interpretation," in Perioperative Assessment of the Maxillofacial Surgery Patient: Problem-based Patient Management, 2018.

[14] G. Huang, S. Liu, L. Van Der Maaten, and K. Q. Weinberger, "CondenseNet: An efficient densenet using learned group convolutions," arXiv. 2017.

[15] S. Ramiz and M. Rajpurkar, "Pulmonary Embolism in Children," Pediatric Clinics of North America. 2018, doi: 10.1016/j.pcl.2018.02.002.

[16] Y. Bengio, "Learning deep architectures for AI," Found. Trends Mach. Learn., 2009, doi: 10.1561/2200000006.

[17] P. Rajpurkar et al., "CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning," arXiv. 2017.

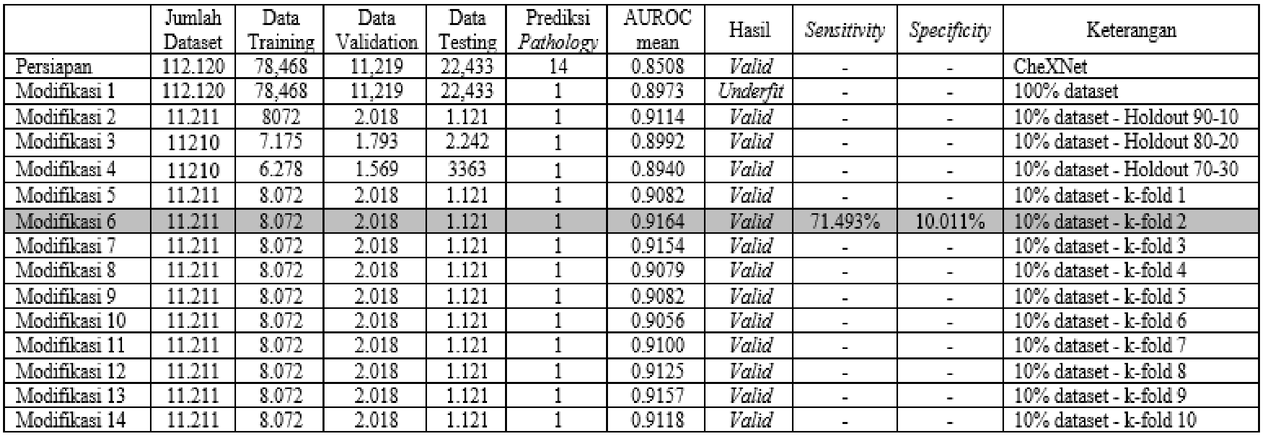
[18] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," 2014, doi: 10.3115/v1/p14-1062.

[19] Y. Zhang, J. Gao, and H. Zhou, "Breeds Classification with Deep Convolutional Neural Network," 2020, doi: 10.1145/3383972.3383975.

[20] C. Neural and N. Accelerator, "ISAAC : A Convolutional Neural Network Accelerator with I n- S itu A nalog A rithmetic in C rossbars," Iscas, 2016.

[21] R. M. PRASMATIO, B. Rahmat, and I. Yuniar, "Deteksi Dan Pengenalan Ikan Menggunakan Algoritma Convolutional Neural Network," J. Inform. dan Sist. Inf., 2020.

Appendix 1. List of model test results



	Jumlah Dataset	Data Training	Data Validation	Data Testing	Prediksi *Pathology*	AUROC mean	Hasil	*Sensitivity*	*Specificity*	Keterangan
Persiapan	112.120	78,468	11,219	22,433	14	0.8508	*Valid*	-	-	CheXNet
Modifikasi 1	112.120	78,468	11,219	22,433	1	0.8973	*Underfit*	-	-	100% dataset
Modifikasi 2	11.211	8072	2.018	1.121	1	0.9114	*Valid*	-	-	10% dataset - Holdout 90-10
Modifikasi 3	11210	7.175	1.793	2.242	1	0.8992	*Valid*	-	-	10% dataset - Holdout 80-20
Modifikasi 4	11210	6.278	1.569	3363	1	0.8940	*Valid*	-	-	10% dataset - Holdout 70-30
Modifikasi 5	11.211	8.072	2.018	1.121	1	0.9082	*Valid*	-	-	10% dataset - k-fold 1
Modifikasi 6	11.211	8.072	2.018	1.121	1	0.9164	*Valid*	71.493%	10.011%	10% dataset - k-fold 2
Modifikasi 7	11.211	8.072	2.018	1.121	1	0.9154	*Valid*	-	-	10% dataset - k-fold 3
Modifikasi 8	11.211	8.072	2.018	1.121	1	0.9079	*Valid*	-	-	10% dataset - k-fold 4
Modifikasi 9	11.211	8.072	2.018	1.121	1	0.9082	*Valid*	-	-	10% dataset - k-fold 5
Modifikasi 10	11.211	8.072	2.018	1.121	1	0.9056	*Valid*	-	-	10% dataset - k-fold 6
Modifikasi 11	11.211	8.072	2.018	1.121	1	0.9100	*Valid*	-	-	10% dataset - k-fold 7
Modifikasi 12	11.211	8.072	2.018	1.121	1	0.9125	*Valid*	-	-	10% dataset - k-fold 8
Modifikasi 13	11.211	8.072	2.018	1.121	1	0.9157	*Valid*	-	-	10% dataset - k-fold 9
Modifikasi 14	11.211	8.072	2.018	1.121	1	0.9118	*Valid*	-	-	10% dataset - k-fold 10

khazanah informatika

Jurnal Ilmu Komputer dan Informatika

Analysis of the Causal Relationship of Body Image Factors in Patients with Cancer

Vita Ari Fatmawati[1], Christantie Effendy[2], Ridho Rahmadi[1*]
[1]Master of Informatics Program
Faculty of Industrial Technology
Universitas Islam Indonesia
Yogyakarta, Indonesia
[2]Department of Medical Surgical Nursing
Faculty of Medicine, Public Health and Nursing
Universitas Gadjah Mada
Yogyakarta, Indonesia
*ridho.rahmadi@uii.ac.id

Abstract-Patients with cancer can potentially experience the negative impacts of treatment. Physical conditions due to illness and therapy can affect the patient's body image. This study aims to find a causal model among body image factors of patients with cancer using the S3C-Latent Method. The measurement of body image of patients with cancer used the BIS questionnaire. One hundred and ninety-nine patients with cancer participated in this study. The results showed the existence of causal relationships between behavior to cognitive factors and duration of illness with reliability scores of 0.8 and 0.6, respectively; from gender to affective factors, illness duration, behavior, and cognitive factors with reliability scores of 0.6, 0.8, 0.65, and 1, respectively. There are also causal relationships from age to affective factors, duration of illness, and cognitive factors with reliability scores of 0.8, 0.7, and 0.9, respectively. The results also showed that affective factors are associated with behavior, cognitive factors, and duration of illness, with reliability scores of 1, 1, and 0.9, respectively. The results showed further the association of cognitive factors and illness duration with a reliability score of 1. We expect that the estimated causal model will serve as a scientific reference for medical experts in developing a better intervention such as treatment.

Keywords: cancer, body image, s3c-latent, causal relationship

Article info: *submitted April 30, 2021, revised June 24, 2021, accepted July 11, 2021*

1. Introduction

Health is among the most important factors in life. Various diseases can be a threat to humans, one of which is cancer. Cancer is caused by the abnormal growth of body cells as a result of mutations and changes in biochemical structure [1]. The International Agency for Research on Cancer (IARC) stated that the global cancer burden will increase by 19.3 million cases and 10 million deaths in 2020. New cancer cases in Indonesia in 2020 are around 396,914 people. Socioeconomic risk factors are the main factors driving the increase in cancer cases in the world.

Treatment in cancer patients can be divided into surgical (surgery) and systemic (chemotherapy, radiotherapy) [2]. Patients with cancer who receive surgical treatment are often faced with temporary or permanent effects that affect their appearance. The impact of treatment can be problems, such as scars or even loss of body parts due to amputation after surgery, sunburn from radiation therapy, and hair loss due to chemotherapy [3]. The patient's physical changes can affect the body image. Negative body image comes from negative thoughts and feelings which leads to decrease in the patient's self-esteem [4]. Body image is a primary factor of self-esteem.

Body image is a description of a person's assessment of physicality, satisfaction, and acceptance of the body. It can be interpreted as a form of estimation and evaluation of individuals on their bodies based on social norms and judgments from others [5]. Society places a high value on the beauty of the human body. Every individual has different perceptions about their own body, it is not uncommon for a person's perception to differ from the standards and expectations of society in general. Body image refers to a state in which a person perceives facts about their body, and whether they are satisfied or dissatisfied with their body. The level of individual satisfaction determines the level of self-confidence and self-esteem [6]. Measuring the body image of patients with cancer can be administered by using the BIS questionnaire. BIS was developed to measure body image in patients with cancer and it consists of 10 assessment items with five positive questions and five negative questions. The BIS test results showed a high reliability value (Cronbach's alpha 0.93). The body image of patients with cancer is influenced by three factors, i.e., affective, behavior, and cognitive [7].

Causal inference has been widely applied to draw causal conclusions based on observational studies, by providing non-randomized observational treatment [8]. Causal modeling is currently commonly used in the fields of bioinformatics, medicine, image processing, sports outcome prediction, risk analysis, and quantum non-locality research [9]but where the ancestry of each copy mirrors that of the original. To every distribution of the observed variables that is compatible with the original causal structure, we assign a family of marginal distributions on certain subsets of the copies that are compatible with the inflated causal structure. It follows that compatibility constraints for the inflation can be translated into compatibility constraints for the original causal structure. Even if the constraints at the level of inflation are weak, such as observable statistical independences implied by disjoint causal ancestry, the translated constraints can be strong. We apply this method to derive new inequalities whose violation by a distribution witnesses that distribution's incompatibility with the causal structure (of which Bell inequalities and Pearl's instrumental inequality are prominent examples. In public health, causal inference has a central role [10], as it attempts to understand causal mechanisms underlying the system, which will be useful in developing intervention.

A causal model can be represented by a directed acyclic graph (DAG) or equivalently a structural equation model (SEM), and with a strong assumption, it can be used to describe the mechanisms that generated the observed data [11]. The causal models represent of the causal networks connecting exposure, effect, confounders, and other variables, that require explicit formulation of the relationships among these factors. As a result, causal models are a useful method for detecting graphically, identify potential causes of bias, and to advise investigators in the design of their data analysis [10].

Stable Specification Search for Cross-Sectional Data (S3C) is an exploratory causal approach that estimates causal models with observed variables. S3C combines a multi-objective optimization approach with stability selection concept to search for stable and parsimonious causal structures. There are two major sections of the S3C procedure. The first phase searches for relevant (stable and parsimonious) causal structures for the entire model complexities. The second part of S3C visualizes those relevant causal structures [12]. The extended version of S3C, that is, S3C-Latent, is developed to model causal relations among latent variables or factors. Latent variable models are commonly used in psychology, mental health studies, and other related areas [18].

2. Methods

This research is a quantitative study with an exploratory method using a cross sectional study design. The stages in this research include literature review, pre-processing data, causal modeling, evaluation, and dissemination. The research stages can be seen in Figure 1.

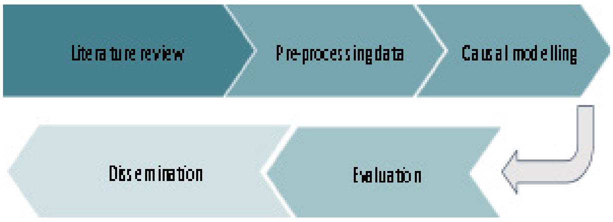


Figure 1. Research stages

a. Literature review

The research stage begins with a study of some relevant previous studies. Body image is a complex construct which describes how an individual views his/her physical appearance. Perceptions, thoughts, feelings, and behaviors about the entire body and its functions are of body image. Changes in body shape as a result of disease or treatment undergoing cancer patients can affect the body image. Treatment of patients with cancer has impact on both physical and psychological conditions [2]. Cancer patients undergoing chemotherapy experience decreased body health, body image disorders, sexual dysfunction, reduced social participation, and decreased work ability compared to the condition before chemotherapy [13].

Age, BMI, and various cancer treatments have been reported as potential risk factors for body image issues of patients with cancer [14]. Family support and marital status affect body image in post-mastectomy patients with cancer [4]. The physical condition of the patient after the mastectomy can reduce the patient's self-confidence. Family support can help patients to deal with self-concept problems. Married patients with breast cancer body image change after surgery, as the patients feel ashamed of their partners and have concern about the risks on their children, while unmarried patients tend to feel worried and ashamed of their physical condition, are afraid of being ostracized and are hard to find a mate.

S3C is designed to model causal relationships between observed variables. S3C is an algorithm for determining causal relationships based on scores that aims to find stable specifications for robust cross-sectional data for a limited sample based on advances in stability selection using subsampling and selection algorithms. S3C combines a structure search process through SEM, NSGA-II, stability selection and visualization processes to present relevant relationships as a causal model. S3C extended to Stable Specification Search for Cross-Sectional Data for latent variables (S3C-Latent). S3C-Latent developed to model causal relationships among latent variables [12]. This study will use S3C-Latent to estimate causal relationships among factors representing the body image.

b. Pre-processing data

The present study uses a dataset from previous study [15] with a span of data collection time in July 2017-February 2018. The dataset consists of 199 patients with cancer. Data collection tools in this study used a demographic data questionnaire and a Body Image Scale (BIS) questionnaire. BIS was developed to measure body image in patients with cancer. In collaboration with the European Organization for Research and Treatment of Cancer (EORTC) Quality of Life Study Group, a 10-item scale was constructed. BIS was tested in a heterogeneous population of 276 British cancer patients. The scale underwent a psychometric testing on 682 breast cancer patients from seven UK clinical trials based on the latest revision. The results of the BIS test showed a high reliability value (Cronbach's alpha 0.93).

Table 1. List of questions in BIS questionnaire

Variables	Statement
Affective	1. Have you been feeling self-conscious about your appearance?
	2. Have you felt less physically attractive as a result of your disease or treatment?
	3. Have you been dissatisfied with your appearance when dressed?
	4. Have you been feeling less feminine/ masculine as a result of your disease or treatment?
Behavior	5. Did you find it difficult to look at yourself naked?
	6. Have you been feeling less sexually attractive as a result of your disease or treatment?
	7. Did you avoid people because of the way you felt about your appearance?
Cognitive	8. Have you been feeling the treatment has left your body less whole?
	9. Have you felt dissatisfied with your body?
	10. Have you been dissatisfied with the appearance of your scar?

BIS consists of 10 assessment items with five positive items and five negative items. BIS used a Likert scale, that is, four alternative answers ranging from the choice of "not at all" (0) to "very much" (3) [7]. The total score obtained from 0 to 30 can be calculated by adding up the scores of the 10 items. A higher score means a higher level of body image disturbance [16]. Body image factors of patients with cancer, are affective, behavior, and cognitive factors. Affective factors are indicated by assessment items 1-4, behavior factors by items 5-7, and cognitive factors by items 8-10 [7].

This study uses demographic characteristics such as age, gender, education level, marital status, and duration of illness. Grouping the characteristics of the respondents is used to determine the distribution and percentage of each variable (see Table 2). Before the computation process, the dataset is checked for missing values, data distribution, data consistency or duplicate data in the dataset, and converted into a new format compatible for computations in R programming.

Table 2. Distribution of respondent characteristics

Respondent characteristics	Patients (n=199) n (%)
Age	
<30	4 (2.0)
30-40	27 (13.6)
41-50	63 (31.7)
51-60	76 (38.2)
>60	29 (14.6)
Gender	
Female	183 (92.0)
Male	16 (8.0)
Marital status	
Married	166 (83.4)
Not married	31 (15.6)
Other	2 (1)
Education level	
No formal education	20 (10.1)
Primary school	90 (45.2)
Junior high school	25 (12.6)
Senior high school	40 (20.1)
Diploma	7 (3.5)
Bachelor	17 (8.5)
Duration of illness	
<3 months	17 (8.5)
3-6 months	61 (30.7)
7-12 months	56 (28.1)
1-24 months	22 (11.1)
>24 months	43 (21.6)

Table 2 summarizes information about demographic variables and medical characteristics among respondents. Almost 38,2% of the cancer patients were age 51-60, 92% are women, 83,4% were married. Most had attended primary school (45.2%), and the most length of illness is between 3-6 months (30.7%).

c. Causal modeling

Causal modeling aims to determine causal relationship among several variables. A causal model can be represented with a diagram of the relationships among independent, control, and dependent variables [17]. The observed variables represented by boxes, the unobserved/ latent variables by circles [18].

The computation in this study is conducted in parallel using a cluster computer with R v4.0.0 as language programming and R package, Stablespec. The CPU server equipped with 80 cores, 250 GB RAM, and 4 GPU, Jupyter GUI and console/terminal.

1) Structural Equation Model

S3C uses SEM to represent causal relationships. SEM is considered as the main language of causal modeling [19]. S3C-Latent employs SEM with latent variables which comprises a structural model and measurement model. The structural model for latent variables reads

$$\boldsymbol{\eta} = \boldsymbol{B\eta} + \boldsymbol{\Gamma\xi} + \boldsymbol{\zeta} \tag{1}$$

Where $\boldsymbol{\eta}$ is an $m \times 1$ vector for the endogenous latent (effect) variables, $\boldsymbol{\xi}$ is an $n \times 1$ vector for exogenous latent (cause) variables, $\boldsymbol{\zeta}$ is an $m \times 1$ vector for disturbance on $\boldsymbol{\eta}$, $\boldsymbol{B}$ is an $m \times m$ matrix for coefficients among $\boldsymbol{\eta}$ and Γ is an $m \times n$ matrix for coefficient among $\boldsymbol{\xi}$. In addition, there are values $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ which represent the covariance matrix of $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$, respectively. It is assumed that E ($\boldsymbol{\eta}$) = E ($\boldsymbol{\xi}$) = E ($\boldsymbol{\zeta}$) = 0, $\boldsymbol{\xi}$ has no correlation with $\boldsymbol{\zeta}$ and ($\boldsymbol{I}$ - $\boldsymbol{B}$) is non-singular.

The measurement model represents the effect from $\boldsymbol{\eta}$ to the observed variable, an $r \times 1$ vector x, and from $\boldsymbol{\zeta}$ to the observed variable, a $q \times 1$ vector y. The measurements model reads

$$\boldsymbol{x} = \boldsymbol{\Lambda_x \xi} + \boldsymbol{\delta} \tag{2}$$

$$\boldsymbol{y} = \boldsymbol{\Lambda_y \eta} + \boldsymbol{\epsilon}, \tag{3}$$

where $r \times n$ matrix $\boldsymbol{\Lambda}_x$ and q x m matrix $\boldsymbol{\Lambda}_y$ contain the structural coefficients that connect the latent variables and indicators, the vectors $r \times 1$ vector $\boldsymbol{\delta}$ and $q \times 1$ vector $\boldsymbol{\epsilon}$ contain errors value on the indicators. In addition, the $r \times r$ matrix $\boldsymbol{\Theta}_\delta$ and a $q \times q$ matrix $\boldsymbol{\Theta}_\epsilon$ are the covariance matrices for $\boldsymbol{\delta}$ and $\boldsymbol{\epsilon}$. The indicator in x and y represent independent questionnaire items and assumed that no causal relation between them.

The next procedure is a model parameter estimation. The model parameters $\boldsymbol{\theta}$ comprises $\boldsymbol{B}$, $\boldsymbol{\Gamma}$, $\boldsymbol{\Lambda}_x$, $\boldsymbol{\Lambda}_y$, $\boldsymbol{\Phi}$, $\boldsymbol{\Psi}$, $\boldsymbol{\Theta}_\delta$, and $\boldsymbol{\Theta}_\epsilon$. SEM procedure estimates a model-implied covariance $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ and assess how it close to covariance matrix **S**. The covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ for SEM with latent variables is a function of the model parameter $\boldsymbol{\theta}$ through

$$\boldsymbol{\Sigma}(\boldsymbol{\theta}) = \begin{bmatrix} \boldsymbol{\Sigma}_{yy}(\boldsymbol{\theta}) & \boldsymbol{\Sigma}_{yx}(\boldsymbol{\theta}) \\ \boldsymbol{\Sigma}_{xy}(\boldsymbol{\theta}) & \boldsymbol{\Sigma}_{xx}(\boldsymbol{\theta}) \end{bmatrix} \tag{4}$$

$$\boldsymbol{\Sigma}_{yy}(\boldsymbol{\theta}) = \boldsymbol{\Lambda}_{\mathrm{y}}(\boldsymbol{I} - \boldsymbol{B})^{-1}(\boldsymbol{\Gamma\Phi\Gamma}' + \boldsymbol{\Psi})$$
$$[(\boldsymbol{I} - \boldsymbol{B})^{-1}]'\, \boldsymbol{\Lambda}_y' + \boldsymbol{\Theta}_\epsilon, \tag{5}$$

$$\boldsymbol{\Lambda}_x \boldsymbol{\Phi\Gamma}'[(\boldsymbol{I} - \boldsymbol{B})^{-1}]'\, \boldsymbol{\Lambda}_y', \tag{6}$$

$$\boldsymbol{\Sigma}_{xx}(\boldsymbol{\theta}) = \boldsymbol{\Lambda}_x \boldsymbol{\Phi\Lambda}_x' + \boldsymbol{\Theta}_\delta\ , \tag{7}$$

where $\boldsymbol{\Sigma}_{yy}(\boldsymbol{\theta})$ is a covariance matrix of the indicator $\boldsymbol{y}$, $\boldsymbol{\Sigma}_{xy}(\boldsymbol{\theta})$ is a covariance matrix of the indicator $\boldsymbol{x}$ and $\boldsymbol{y}$, and $\boldsymbol{\Sigma}_{xx}(\boldsymbol{\theta})$ as a covariance matrix of the indicator $\boldsymbol{x}$.

A DAG, and therefore SEM, has its corresponding Markov equivalence class or so called a partially completed DAG (CPDAG) [20]. This means that any distribution of probabilities entailed by a DAG belonging to a particular CPDAG may also be obtained from other DAG belonging to the same CPDAG.

SEM with latent variables has several additional identification conditions. First, there are at least three or more indicators for each latent variable. Second, each row $\boldsymbol{\Lambda}_x$ and $\boldsymbol{\Lambda}_y$ has one non-zero element. Third, each latent variable is scaled. Fourth, the value of $\boldsymbol{\Theta}_\delta$ is diagonal [12].

After the SEM identification process passes, the model parameter $\boldsymbol{\theta}$ can be estimated by finding the maximum value using likelihood procedure to find the smallest cost function value. The smaller the value of the cost function, the closer the model with the actual data [12].

$$\widehat{\boldsymbol{\theta}} = \underset{\theta}{argmin}\ F_{ML}(\boldsymbol{\theta}), \tag{8}$$

$$F_{ML}(\boldsymbol{\theta}) = \log|\sum(\boldsymbol{\theta})| + \mathrm{Tr}\,\{\boldsymbol{S\Sigma}^{-1}(\boldsymbol{\theta})\} -$$
$$\log|\boldsymbol{S}| - p. \tag{9}$$

Where $p = r + q$ the number of observed variables, and $\boldsymbol{S}$ is the covariance matrix p x p of observed variables.

2) Non-dominated Sorting Genetic Algorithm II (NSGA-II) and Stability Selection

The selected models are then processed with NSGA-II. NSGA-II mimics the evolution cycle, by manipulating the best model from the previous selection. The best models of the first generation are manipulated (using operators called crossover and mutation methods) and are used to produce the second-generation model which is expected to be better than those of the first. This process is repeated until many generations. The crossover ensures variability of models for the next generation and the mutation is carried out to ensure that the model optimization will not get stuck in the local optima.

The model must be able to anticipate data fluctuations, this requires a stability selection process. Stability selection applies an iterative variable selection algorithm to a randomly drawn subset of half of the original data. The variable is selected when meets the predetermined threshold [12]. The best models from different subsets are collected, then combined and calculated using the concept of stability selection. The models are categorized based on their complexity, then the stability is calculated per pair of variables. Per pair of variables are calculated two types of stability. The first stability is all kinds of relationships, looking for its edge stability value, and the second by the causal path stability.

Final stage, drawing the plot per pair of the variables. The causal model based on the predetermined stability threshold (the stability threshold is generally 0.6) and the complexity threshold value using the BIC Score. Relevant structures are connected to each other based on relevant edges and their direction based on causal path stability.

3) Stable Specification Search for Cross Sectional Data (S3C)

S3C looks for causal models based on score. S3C combines the concept of stability selection and multi-object optimization to find a stable and simple model structure for the observed variables. S3C method can be seen in Figure 2 [12].

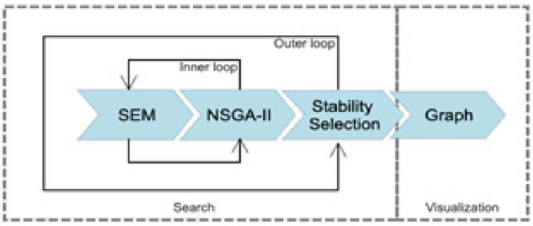
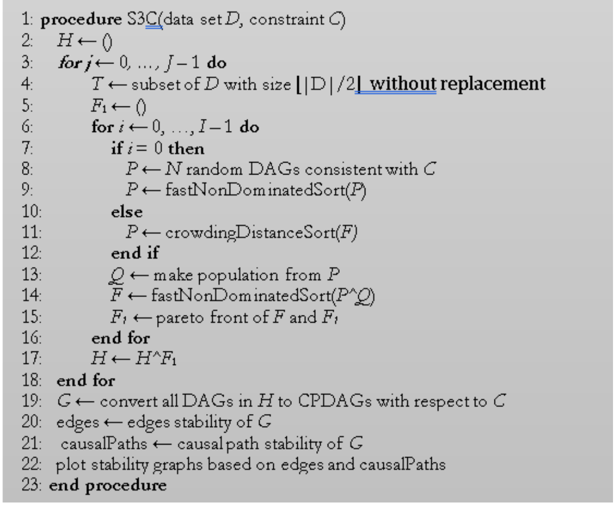


Figure 2. S3C method

SC3 consists of two phases, the optimal model search phase and the visualization phase. The search phase is an iterative procedure that requires an outer loop and an inner loop to find relevant edges and causal paths between two variables using SEM, NSGA-II and stability selection [12]. The relevant relationships that are displayed in the visualization phase are called causal models. The S3C pseudocode can be seen in Figure 3 [12].



```
1: procedure S3C(data set $D$, constraint $C$)
2:   $H \leftarrow ()$
3:   for $j \leftarrow 0, \ldots, J-1$ do
4:       $T \leftarrow$ subset of $D$ with size $\lfloor |D|/2 \rfloor$ without replacement
5:       $F_1 \leftarrow ()$
6:       for $i \leftarrow 0, \ldots, I-1$ do
7:           if $i = 0$ then
8:               $P \leftarrow N$ random DAGs consistent with $C$
9:               $P \leftarrow$ fastNonDominatedSort($P$)
10:          else
11:              $P \leftarrow$ crowdingDistanceSort($F$)
12:          end if
13:          $Q \leftarrow$ make population from $P$
14:          $F \leftarrow$ fastNonDominatedSort($P$^$Q$)
15:          $F_1 \leftarrow$ pareto front of $F$ and $F_1$
16:      end for
17:      $H \leftarrow H$^$F_1$
18:  end for
19:  $G \leftarrow$ convert all DAGs in $H$ to CPDAGs with respect to $C$
20:  edges $\leftarrow$ edges stability of $G$
21:  causalPaths $\leftarrow$ causal path stability of $G$
22:  plot stability graphs based on edges and causalPaths
23: end procedure
```

Figure 3. The pseudocode of S3C

The S3C procedure is divided into an inner loop and an outer loop. The inner loop used to find the pareto front with the NSGA-II implementation, while the outer loop created a sample from a subset of data. Pareto fronts were converted into CPDAG which was used to calculate the edge and causal path stability graphs.

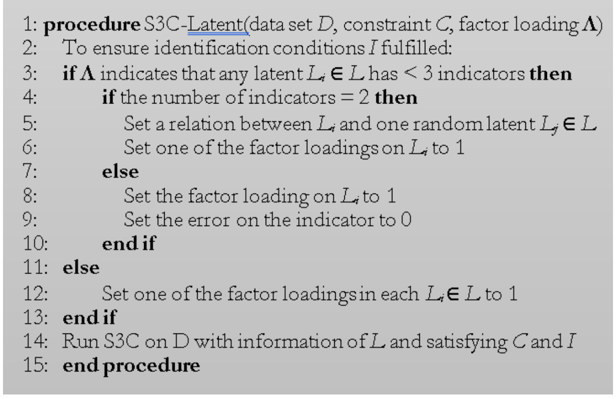
Lines 6-16 interpreted an inner loop begins by randomly generating a population P of size N or using crowding distance sorting to get a previous population (Lines 7-12). A binary vector {0,1} is a representation of the model, and denoted by some arc $X \rightarrow Y$. A new population of Q is the result of P manipulation using binary tournament selection, one-point crossover, and one-bit flip mutation. The best model from the selection placed in a mating pool M_{pool} with selection scheme of taking the N model twice from P. Two models from M_{pool} are taken by one-point crossover and after the crossover point, data is swapped in the middle. Mutation of one-bit flip flips each bit according to a predetermined rate. The combination of P and Q using fast non-dominated sorting in Line 14 resulted a set of model fronts F. The Pareto front in F_1 updated by Line 15.

Lines 3-18 interpreted an outer loop begins by randomly samples from D to a subset T with size $\lfloor|D|/2\rfloor$ (Line 4), Lines 6-16 run the inner loop with I times to get a Pareto front, Line 17 places it in H. H contains J Pareto fronts after J iterations is done.

J Pareto fronts in H converts DAGs into CPDAG (Lines 19-22) and then computes the edge and causal path stability graphs. The main output of S3C is the stability graphs which visualizes as a graph with nodes and edges [12].

4) Stable Specification Search for Cross Sectional Data for Latent Variables (S3C-Latent)

S3C-Latent was extended from S3C. The difference between S3C and S3C-Latent, S3C uses observed variables while S3C-Latent uses latent/unobserved variables. Pseudocode S3C-Latent was developed from S3C for computing latent variables. The S3C-Latent pseudocode can be seen in Figure 4 [12].



```
1: procedure S3C-Latent(data set $D$, constraint $C$, factor loading $\mathbf{\Lambda}$)
2:   To ensure identification conditions $I$ fulfilled:
3:   if $\mathbf{\Lambda}$ indicates that any latent $L_i \in L$ has $< 3$ indicators then
4:       if the number of indicators $= 2$ then
5:           Set a relation between $L_i$ and one random latent $L_j \in L$
6:           Set one of the factor loadings on $L_i$ to 1
7:       else
8:           Set the factor loading on $L_i$ to 1
9:           Set the error on the indicator to 0
10:      end if
11:  else
12:      Set one of the factor loadings in each $L_i \in L$ to 1
13:  end if
14:  Run S3C on D with information of $L$ and satisfying $C$ and $I$
15:  end procedure
```

Figure 4. The pseudocode of S3C-latent

D is the data set, L is a set of n latent variables, $\mathbf{\Lambda}$ is a matrix of factor loadings, and C is a prior knowledge. Line 3 to confirm whether there is a latent variable $L_i \in L$ which has less than 3 indicators. If found, then for each L_i, Line 4 will be executed or vice versa Line 12 will be executed. Line 4 checks the number of indicators L_i is 2 or 1. If a case of 2 indicators is found, SC3-Latent establishes the relationship of the latent variable L_i and random latent variable $L_j \in L$ (relates the latent variable L_i to L_j, where L_i can be cause or effect), and set one of the loading factors to 1 (Lines 5 and 6). If the indicator is 1, S3C-Latent will set the loading factor L_i to 1 and set the error on the indicator to 0 (Lines 8 and 9). When all latent variables have at least 3 indicators then Line 12 exists. Line 14 runs S3C on the data set (*D*) with the latent variable information from *L*, and satisfies any constraint in *C* and the model identification conditions in *I*. S3C-Latent ensures that all SEMs that are generated and refined are compatible with the previous specified in *C* by meeting constrains in *C* (if any).

a. Evaluation

The model obtained in this study will be evaluated by experts in related field such as oncology nurses. The evaluation process uses a questionnaire. The evaluation will be conducted by online survey.

b. Dissemination

The final stage of this study aims to apply the causal model into an interactive website based. The application built with a shiny package in R. The application is used to disseminate the model to a wider realm, especially for medical personnel who need information related to body image in patients with cancer.

3. Results

The computation stage begins with determining the parameter settings. The parameters in this study are the subset parameter (*S*), the number of iterations (*I*), the number of models evaluated (*P*), the crossover probability (*C*), and the mutation probability (*M*). This study uses the amount of data n = 199 with the parameters were set as follows: *S* = 150, *I* = 50, *P* = 130, *C* = 0.45, *M* = 0.01. Furthermore, added constraints, set gender and age did not cause body image factors in patients with cancer as constraints. Stability graph can be seen in Figure 5.

The stability graph consists of three dotted lines in blue, green, and red. The blue line (edge stability) depicts the relationship between pairs of variables regardless of direction, while the green and red lines represent the stability of the causal path with one length or any length that describes the causal relationship from one variable to another [12]. Based on stability graphs, behavior factors influenced cognitive factors and duration of illness. Gender has causal relations with affective, behavior, cognitive factors, and duration of illness. Age has causal relations with affective, cognitive factors, and duration of illness. The results also showed that affective associated with behavior, cognitive factors, duration of illness, and cognitive associated with duration of illness.

S3C-Latent is an exploratory score-based causal method that uses multi-objective optimization and stability selection to find robust causal relations (stable and simple) among latent variables in a structural model. The search for stable and parsimonious structural models used two thresholds. The first threshold was the boundary of selection probability π_{sel}, in this research used π_{sel} = 0.6 [21], it means that all causal relationships with edge stability or causal path stability that exceed this value is considered stable. The second threshold was the boundary of complexity π_{bic} that used to control overfitting [12], in this research used π_{bic} = 12, it means that all causal relationships with edge stability or causal path stability lower than π_{bic} are considered parsimonious.

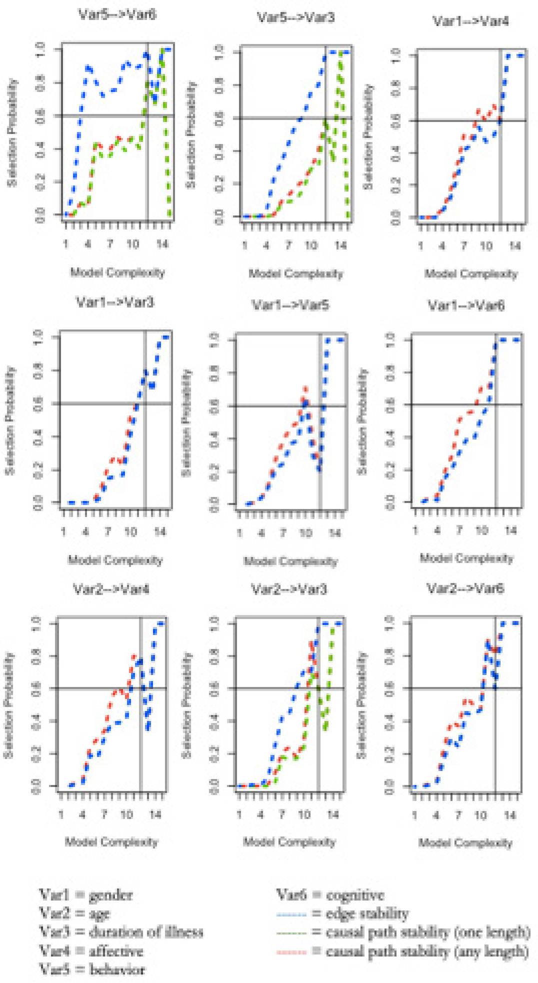
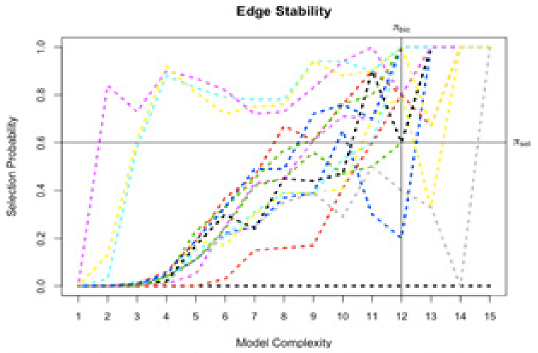
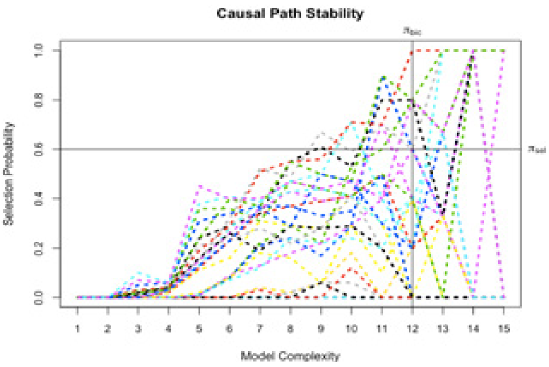


Figure 5. Stability Graph



Each line in edge stability graph represents an edge between a pair of variables

Figure 6. Edge Stability Graph



Each line in causal path stability graph represents a causal path with any length from a variable to another variable

Figure 7. Causal Path Stability Graph

Based in Figure 6 and 7, the edge stability and causal path stability graphs used πsel = 0.6 and πbic = 12. The edge stability graph showed there were 13 relevant edges on the top-left area. Based on causal path stability graph showed that there were nine relevant causal paths on the top-left area of the causal path stability graph.

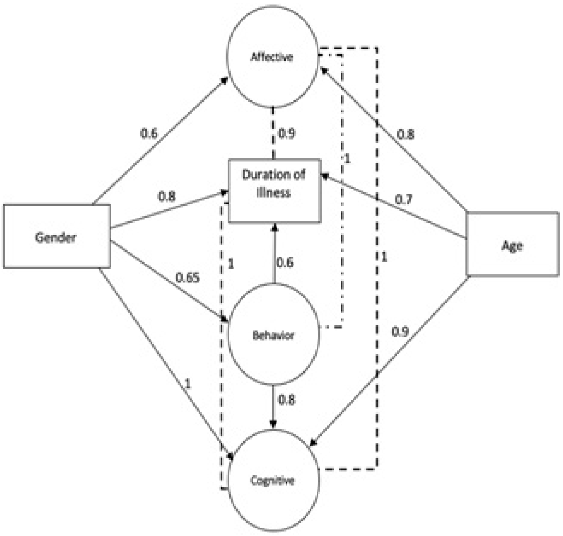
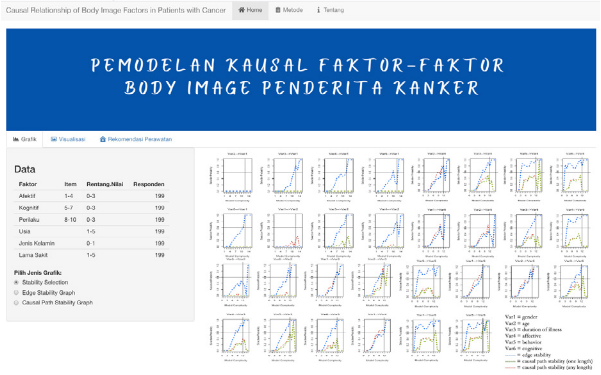


Figure 8. Causal model of body image patients with cancer estimated by S3C-latent algorithm

Causal modeling of body image factors in patients with cancer showed in Figure 8. There are causal relationships between behavior with cognitive factors and duration of illness with reliability scores of 0.8 and 0.6. The causal relations between gender and affective, duration of illness, behavior, cognitive factors with reliability scores of 0.6, 0.8, 0.65, and 1, respectively. Age relations with affective, duration of illness, and cognitive with reliability scores of 0.8, 0.7, and 0.9, respectively. The results also show that affective is associated with behavior factors, cognitive factors, and duration of illness with reliability scores of 1,1, and 0.9, respectively. Cognitive factors and duration of illness are associated with reliability score of 1.

Based on the modeling results obtained, two kinds of relation, causal relationship, and association. All causal relationships are associational, but not all associational relationships are causal, other than that correlation is not the same as causal [22]. Association is an improvement of correlation relationship.

The next stage, we implemented the results into the R Shiny App. R Shiny framework is a package from RStudio to build interactive web application with R. The application displays information about the results of causal modeling of the body image factors in patients with cancer. The website is designed to display an explanation of the data, method, graphs, visualizations, and treatment recommendations, see Figure 9.


Causal Relationship of Body Image Factors in Patients with Cancer | Home | Metode | Tentang

PEMODELAN KAUSAL FAKTOR-FAKTOR
BODY IMAGE PENDERITA KANKER

Grafik | Visualisasi | Rekomendasi Perawatan

Data

Faktor	Item	Rentang.Nilai	Responden
Afektif	1-4	0-3	199
Kognitif	5-7	0-3	199
Perilaku	8-10	0-3	199
Usia		1-5	199
Jenis Kelamin		0-1	199
Lama Sakit		1-5	199

Pilih Jenis Grafik:
● Stability Selection
○ Edge Stability Graph
○ Causal Path Stability Graph

Var1 = gender
Var2 = age
Var3 = duration of illness
Var4 = affective
Var5 = behavior
Var6 = cognitive
= edge stability
= causal path stability (one length)
= causal path stability (any length)

Figure 9. Graphs menu

4. Discussion

Based on the results obtained, there is a causal relationship between behavior and cognitive with a reliability score of 0.8. This result in line with research [23] that nurses providing caring behavior to patients, the patients will be motivated, this is commonly called a cognitive process. Caring behavior by nurses can be a form of social support that can increase the patient's life expectancy.

The causal relationship between behavior and duration of illness with reliability score of 0.6. This research [24] found that wound healing after surgery and injury recovery are necessary for the patient. A poor wound healing process increases the risk of wound complications and infection, prolongs hospital stay, creates discomfort, and hinders daily activities. Psychological stress and other behavioral factors can affect the wound healing process.

The causal relationships between gender with affective and cognitive with reliability scores of 0.6 and 1, respectively in line with this research [25] that women are more mentally emotional than men. Women diagnosed with cervical cancer face many difficulties in accepting the fact that they are diagnosed with cancer. Their anxiety increased when imagining life changes and the side effects of treatment received. This would reduce a person's cognitive capacity in solving problems. Patients who experience anxiety close themselves off from other people. While those who can accept will be helped in the healing process. Encouragement and family support is one of the important factors in increasing women's awareness in preventing cervical cancer [26].

The causal relationships between gender with behavior and duration of illness with reliability scores of 0.65, 0.8, respectively has confirmed with this research [27]understanding that the disease is incurable and the advanced stage of the disease. To evaluate gender differences in patients' reports of discussions of life expectancy with oncology providers and its effect on differences in illness understanding. Methods Coping with Cancer 2 patients (N = 68 that female patients with advanced cancer had a higher understanding of the disease than male patients. Female patients had more communication time with oncology providers about life expectancy than male patients. Efforts are needed to improve communication, especially for male patients in order to gain more understanding about their disease in order to get strong information to make medical decisions for patients at the end of their lives so that they can increase their life expectancy.

The causal relationship between age and affective with reliability score of 0.8 according to [28] found that women under the age of 40-60 years have higher body image dissatisfaction. Physical attractiveness in young women had a positive correlation with levels of happiness and self-esteem and had a negative correlation with levels of neurotics or anxiety. Higher bodies rate for young adult women associated with levels of self-satisfaction, respect from others, and sexual quality.

The causal relationship between age and duration of illness with reliability score of 0.7 in line with this study [29] that respondents between ages of 16-65 years had the most cancer (78.4%). The older person has the lower body's immune level. Decreased body immunity will make it easier for cancer to multiply. Age influenced by an unhealthy lifestyle, irregular eating patterns, stress due to heavy workloads, smoking habits, and exposure to cigarette smoke when young can be the cause of cancer detected when old.

The causal relationship between age and cognitive with reliability score of 0.9 in line with the research [30] that cognition crucial for support functional independence as we get older, this includes the ability to live independently, financial management, how to take the right medicine, and how to drive safely. Cognition provides an important role for humans to communicate effectively, including processing and integrating sensory information, and responding appropriately to others. Cognitive abilities often decline with age. Changes in cognition resulting from the normal aging process decrease performance of cognitive tasks that require rapid processing of decision-making, including measures of processing speed, working memory, and executive cognitive function.

The association relationships between affective with behavior and cognitive with reliability scores of 1, 1, respectively has confirmed [31]perspective-taking is closely linked to human empathy, and like empathy, perspective-taking is commonly subdivided into cognitive and affective components. While the two components of empathy have been frequently compared, the differences between cognitive and affective perspective-taking have been under-investigated in the cognitive neuroscience literature to date. Here, we define cognitive perspective-taking as the ability to infer an agent's thoughts or beliefs, and affective perspective-taking as the ability to infer an agent's feelings or emotions. In this paper, we review data from functional imaging studies in healthy adults as well as behavioral and structural imaging studies in patients with behavioral variant frontotemporal dementia in order to determine if there are distinct neural correlates for cognitive and affective perspective-taking. Data suggest that there are both shared and non-shared cognitive and anatomic substrates. For example, while both types of perspective-taking engage regions such as the temporoparietal junction, precuneus, and temporal poles, only affective perspective-taking engages regions within the limbic system and basal ganglia. Differences are also observed in prefrontal cortex: while affective perspective-taking engages ventromedial prefrontal cortex, cognitive perspective-taking engages dorsomedial prefrontal cortex and dorsolateral prefrontal cortex (DLPFC that affective is the ability to draw conclusions from the emotions and feelings of others. Affective is closely related to cognitive empathy. Cognitive empathy is the ability to model the emotions of other agents. Affective empathy results from a combination of cognitive and emotional empathy. Cognitive responses relevant in the adaptation process, cognitive factors can affect an event that can cause stress, determine the coping to be used, emotional reactions, physiology, behavior, and social. Behavioral responses represent emotional and physiological responses as a result of cognitive analysis in dealing with stressful conditions [32].

The association relationship between affective and duration of illness with reliability score of 0.9 in line with [33], patient with low levels of optimism tend to be less able to withstand the negative effects of treatment and susceptible to anxiety and depression. Treatment of cancer patients is focused on physical health, so psychological health is often neglected. Psychological health as a main role in the healing process, such as patient optimism in undergoing treatment.

The association relationship between cognitive and duration of illness with reliability score of 1 in line with [34]evaluated them for in-hospital delirium, and assessed global cognition and executive function 3 and 12 months after discharge with the use of the Repeatable Battery for the Assessment of Neuropsychological Status (population age-adjusted mean [±SD] score, 100±15, with lower values indicating worse global cognition that patients in medical and surgical ICUs had potential risk for long-term cognitive disability. At three and twelve months, a longer duration of delirium in hospital was associated with lower global cognition and executive function levels.

5. Conclusion

Causal modeling is currently commonly used in many fields such as bioinformatics, medicine, image processing, sports outcome prediction, risk analysis, and quantum non-locality research. We have conducted causal modeling of body image factors in patients with cancer using S3C-Latent. The resulted model showed that there are causal relationships between behavior with cognitive factors and duration of illness. Gender has causal relationships with affective factors, illness duration, behavior, and cognitive factors. Age has causal relationships with affective factors, illness duration, and cognitive factors. Affective factors are associated with behavior, cognitive factors, illness duration, while cognitive factors are associated with illness duration.

Reference

[1] C. A. Wijaya and M. Muchtaridi, "Pengobatan Kanker Melalui Metode Gen Terapi," *Farmaka*, vol. 15, no. 1, pp. 53–68, 2017.

[2] P. Nova and E. N. Sumintarddja, "Peran BRIEF CBT Terhadap Tingkat Depresi dan Masalah Body Image Pasien Kanker Payudara Dewasa Muda," *J. Ilm. Psikol. MANASA*, vol. 5, no. 2, pp. 103–113, 2016.

[3] H. C. Melissant *et al.*, "A systematic review of the measurement properties of the Body Image Scale (BIS) in cancer patients," *Support. Care Cancer*, vol. 26, no. 6, pp. 1715–1726, 2018, doi: 10.1007/s00520-018-4145-x.

[4] Sriwahyuningsih, Dahrianis, and M. Askar, "Faktor yang berhubungan dengan gangguan citra tubuh (body image) pada pasien post pperasi mastektomi di RSUP dr . Wahidin," *STIKES Nani Hasanuddin Makassar*, vol. 1, no. 3, pp. 1–6, 2012.

[5] L. A. Rozika and N. Ramdhani, "Hubungan antara harga diri dan body image dengan online self-presentation pada pengguna instagram," *Gadjah Mada J. Psychol.*, vol. 2, no. 3, p. 172, 2018, doi: 10.22146/gamajop.36941.

[6] G. K. Tiwari and S. Kumar, "Psychology and body image : a review," *Shodh Prerak*, no. January, 2015.

[7] P. Hopwood, I. Fletcher, A. Lee, and S. Al Ghazal, "A body image scale for use with cancer patients," *Eur. J. Cancer*, vol. 37, no. 2, pp. 189–197, 2001, doi: 10.1016/S0959-8049(00)00353-1.

[8] W. Luo, W. Wu, and Y. Zhu, "Learning Heterogeneity in Causal Inference Using Sufficient Dimension Reduction," *J. Causal Inference*, vol. 7, no. 1, 2019, doi: 10.1515/jci-2018-0015.

[9] E. Wolfe, R. W. Spekkens, and T. Fritz, "The Inflation Technique for Causal Inference with Latent Variables," *J. Causal Inference*, pp. 70–91, 2019, doi: 10.1515/jci-2017-0020.

[10] T. A. Glass, S. N. Goodman, M. A. Hernán, and J. M. Samet, "Causal inference in public health," *Annu. Rev. Public Health*, vol. 34, pp. 61–75, 2013, doi: 10.1146/annurev-publhealth-031811-124606.

[11] M. L. Petersen and M. J. Van Der Laan, "Causal models and learning from data: Integrating causal modeling and statistical estimation," *Epidemiology*, vol. 25, no. 3, pp. 418–426, 2014, doi: 10.1097/EDE.0000000000000078.

[12] R. Rahmadi, "Finding stable causal structures from clinical data," Radboud Universiteit Nijmegen, 2019.

[13] Z.-P. Ai, X.-L. Gao, J.-F. Li, J.-R. Zhou, and Y.-F. Wu, "Changing trends and influencing factors of the quality of life of chemotherapy patients with breast cancer," *Chinese Nurs. Res.*, vol. 4, no. 1, pp. 18–23, 2017, doi: 10.1016/j.cnre.2017.03.006.

[14] D. E. Fingeret, M. C., Teo, I., & Epner, "Managing body image difficulties of adult cancer patients: lessons from available research," *Cancer*, vol. 120, no. 5, pp. 633–641, 2014, doi: 10.1002/cncr.28469.Managing.

[15] S. S. Yatmi Tri, Effendy Christantie, "Gambaran diri dan kualitas hidup pasien kanker payudara," Gadjah Mada, 2018.

[16] S. A. Bahrami, M., Mohamadirizi, M., Mohamadirizi, S., & Hosseini, "Evaluation of body image in cancer patients and its association with clinical variables," *J. Educ. Health Promot.*, vol. 6, no. 81, 2017, doi: 10.4103/jehp.jehp_4_15.

[17] J. M. Youngblut, "A consumer's guide to causal modeling: Part I," *J. Pediatr. Nurs.*, vol. 9, no. 4, pp. 268–271, 1994.

[18] J. M. Youngblut, "A consumer's guide to causal modeling: Part II," *J. Pediatr. Nurs.*, vol. 9, no. 6, pp. 409–413, 1994, [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3624763/pdf/nihms412728.pdf.

[19] P. Judea, *Causality: models, reasoning and inference*. Cambridge: Cambridge University Press, 2000.

[20] D. M. Chickering, "Learning Equivalence Classes of Bayesian-Network Structures," *J. Mach. Learn. Res.*, vol. 2, no. 3, pp. 445–498, 2002, doi: 10.1162/153244302760200696.

[21] N. Meinshausen and P. Bühlmann, "Stability selection," *J. R. Stat. Soc. Ser. B Stat. Methodol.*, vol. 72, no. 4, pp. 417–473, 2010, doi: 10.1111/j.1467-9868.2010.00740.x.

[22] S. D. Stovitz, E. Verhagen, and I. Shrier, "Distinguishing between causal and non-causal associations: implications for sports medicine clinicians," *Br. J. Sports Med.*, vol. 53, no. 7, pp. 398 LP – 399, Apr. 2019, doi: 10.1136/bjsports-2017-098520.

[23] S. P. S. Sulisno Madya, "Artikel asli," *Media Med. Muda*, vol. 1, no. Januari-April, pp. 7–12, 2016,

[Online]. Available: https://ejournal2.undip.ac.id/index.php/mmm/article/viewFile/2545/1527.

[24] P. J. et al Gouin, "NIH Public Access," *NIH Public Access*, vol. 31, no. 1, pp. 81–93, 2011, doi: 10.1016/j.iac.2010.09.010.The.

[25] M. Ubando, "Gender differences in intimacy, emotional expressivity, and relationship satisfaction," *Pepperdine J. Commun. Res.*, vol. 4, no. 13, pp. 19–29, 2016, [Online]. Available: http://digitalcommons.pepperdine.edu/pjcr%5Cnhttp://digitalcommons.pepperdine.edu/pjcr/vol4/iss1/13.

[26] D. Susilawati, "Hubungan Antara Dukungan Keluarga dengan Tingkat Kecemasan Penderita Kanker Serviks Paliatif," *J. Keperawatan Indones.*, vol. 4, no. 2, pp. 87–99, 2013, [Online]. Available: http://ejournal.umm.ac.id/index.php/keperawatan/issue/view/226/showToc.

[27] K. Fletcher *et al.*, "Gender differences in the evolution of illness understanding among patients with advanced cancer," *J. Support. Oncol.*, vol. 11, no. 3, pp. 126–132, 2013, doi: 10.12788/j.suponc.0007.

[28] A. Melliana, *Menjelajah tubuh perempuan dan mitos kecantikan*. Yogyakarta: LKiS Yogyakarta, 2006.

[29] N. Wardana and R. Ernawati, "Hubungan Usia dan Aktivitas Fisik dengan Jenis Kanker di Ruang Kemoterapi RSUD Abdul Wahab Sjahranie Samarinda," *Borneo Student Res.*, no. 2018, pp. 159–165, 2019, [Online]. Available: http://journals.umkt.ac.id/index.php/bsr/article/view/950.

[30] D. L. Murman, "The Impact of Age on Cognition," *Semin. Hear.*, vol. 36, no. 3, pp. 111–121, 2015, doi: 10.1055/s-0035-1555115.

[31] M. L. Healey and M. Grossman, "Cognitive and affective perspective-taking: Evidence for shared and dissociable anatomical substrates," *Front. Neurol.*, vol. 9, no. JUN, pp. 1–8, 2018, doi: 10.3389/fneur.2018.00491.

[32] S. Nyumirah, "Pengaruh Terapi Perilaku Kognitif Terhadap Kemampuan Interaksi Sosial Klien Isolasi Sosial di RSJ Dr. Amino Gondhohutomo Semarang," 2012.

[33] R. Saniatuzzulfa and S. Retnowati, "Program " Pasien PANDAI " untuk Meningkatkan Optimisme Pasien Kanker," *Gadjah Mada J. Prof. Psychol.*, vol. 1, no. 3, pp. 163–172, 2015.

[34] P. P. Pandharipande *et al.*, "Long-term cognitive impairment after critical illness.," *N. Engl. J. Med.*, vol. 369, no. 14, pp. 1306–1316, Oct. 2013, doi: 10.1056/NEJMoa1301372.

khazanah informatika

Jurnal Ilmu Komputer dan Informatika

Blood Glucose Prediction Using Convolutional Long Short-Term Memory Algorithms

Redy Indrawan [1*], Siti Saadah [1], Prasti Eko Yunanto [2]
Informatics Study Program
Telkom University
Bandung
[2]Informatics Department
Telkom University
Bandung
*redy.indrawan@gmail.com, gppras@telkomuniversity.ac.id

Abstract-Diabetes Mellitus is one of the preeminent causes of death to date. Effective procedures are necessary to prevent diabetes and avoid complications that may cause early death. A common approach is to control patient blood glucose, which necessitates a periodic measurement of blood glucose concentration. This study developed a blood glucose prediction system using a convolutional long short-term memory (Conv-LSTM) algorithm. Conv-LSTM is a variation of LSTM algorithms that are suitable for use in time series problems. Conv-LSTM overcomes the lack in the LSTM algorithm because the latter algorithm cannot access the content of previous memory cells when its output gate has closed. We tested the algorithm and varied the experiment to check the effect of the cross-validation ratio between 70:30 and 80:20. The study indicates that the cross-validation using a ratio of 70:30 data split is more stable compared to one with 80:20 data split. The best result shows a measure of 21.44 in RMSE and 8.73 in MAE. With the application of conv-LSTM using correct parameters and selected data split, our experiment attains accuracy comparable to the regular LSTM.

Keywords: diabetes mellitus, blood glucose, recurrent neural network, long short-term memory

Article info: *submitted May 30, 2021, revised July 21, 2021, accepted August 19, 2021*

1. Introduction

Diabetes is a chronic disease with a gradual growth apparent through an increase in blood glucose. Diabetes is a disease that makes a poor impact on people's lives around the world. According to the World Health Organization (WHO), around 422 million adults live with diabetes in 2014 globally, up from 108 million in 1980. The rise in diabetes prevalence has a relation with the increasing number of people with the condition. Deaths due to diabetes were approximately 3.7 million in 2012, which included 1.5 million from the disease and 2.2 million from its complications [1].

There are two types of diabetes categorized as type 1 and type 2. Early diagnosis with diabetes can help prevent or minimize its complications. A systematic approach may prevent diabetes, its complications, and hence premature death [1]. Treatment can be ineffective if the disease is too severe.

The cause of each type is different. The root cause of type 1 diabetes is currently unidentifiable despite medical advancement. In other words, there are no cures. Therefore, predictions are made to create awareness and strengthen patients' alertness to the underlying problem. Type 2 diabetes stems from the body's lack of efficacy in the usage of insulin. Diabetes in any shape or form could be detrimental to one's health and quality of life. The probability of this circumstance happening is higher if the disease is left undiagnosed and untreated. The level of blood glucose is a key indicator in probing diabetes. For that reason, the prediction of blood glucose is important to carry out.

Various studies have been carried out to predict blood glucose with machine learning such as [2], [3], [4] including two studies using LSTM by [5] and [6]. Previous findings and this study focused on diabetes type 1, but the results are also applicable to other types of diabetes. The study of blood glucose prediction written by

[5] used the LSTM network, which consists of a singular LSTM layer, bi-directional LSTM layer, and several other fully connected layers. The latter study yielded better results compared to the ARIMA and SVR methods with Root Mean Square Error (RMSE) at the value of 21.747. Research for blood glucose prediction written by [6] states that LSTM-NN is superior with an RMSE value of 12.38 compared to autoregressive and LSTM.

The Long Short-Term Memory (LSTM) of the RNN can be used to predict blood glucose. This method is often used to predict other problems [6]. The memory structure of LSTM can capture and store complex data patterns. LSTM enables the neural network to perform tasks that were previously impossible [7]. Although LSTM has good performances in solving time series problems, it faces a problem where it can't access the content of its previous memory cell when its output gate is closed [8]. Convolutional LSTM (conv-LSTM) is a variation of LSTM that can solve the problem [9]. Conv-LSTM is a gate-based memory system that has a connection between the previous memory content and the gate. It allows the gate to access the previous memory content.

The performance of conv-LSTM outperforms the traditional LSTM, CNN-LSTM, and CNN in previous research [9]. In research studies [18], conv-LSTM has had good results compared to other methods in in diabetes classification problems. Therefore conv-LSTM was chosen and applied to predict blood glucose in the continuous glucose monitoring (CGM) time series data in this study.

2. Methods

a. LSTM Network

Long short-term memory (LSTM) is a form of recurrent neural network (RNN) that stores long-term information. It was developed by Johannes Hochreiter and Martin Schmidhuber in 1997 [10]. LSTM was designed to answer the problem of long-term dependency on RNN [11].

The memory cell is the main component of LSTM. It coordinates with the three gate units, among them are the input gate, output gate and forget gate. The units can be used for various operations such as memory relay, write, and reset [7].

The output gate can block the output from the memory cell and sigmoidal nonlinearity exists in all gates [7]. Since the state unit can act as an additional input to other gating units, the LSTM architecture can easily solve the long-term dependency issues. [12]. Due to the blockage at the output gate, LSTM cannot access previous memory cell contents [13].

b. Convolutional LSTM Network

Convolutional LSTM allows the gate to take advantage of the previously stored content memory even after the output gate is blocked. The extra connection between each gate and the previous memory content can be added to solve the problem [9]. Conv-LSTM also has gates like LSTM, input gate , forget gate , and output gate . The illustration of conv-LSTM is shown in Figure *1*.

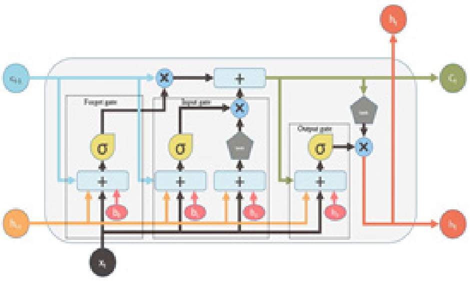


Figure 1 Conv-LSTM Structure [9]

The conv-LSTM implementation takes the previous memory content of the gate as input, , which makes the previous memory cells usable even when the output gate is closed. This connection ensures the earliest input impact even when the input sequence is long. The implementation conv-LSTM is as follows [9]:

$$f = \sigma_g (w_f x_t + U_f h_{t-1} + u_f c_{t-1} + b_f) \quad (1)$$

$$i_t = \sigma_g (w_i x_t + u_i h_{t-1} + u_i c_{t-1} + b_i) \quad (2)$$

$$c_t = f_t c_{t-1} + i_t \sigma_h (w_c x_t + u_c h_{t-1} + b_c) \quad (3)$$

$$o_t = \sigma_g (w_o x_t + u_o h_{t-1} + u_o c_{t-1} + b_o \quad (4)$$

$$h_t = o_t \, tanh \, (c_t) \quad (5)$$

Here, the symbols and denote the sigmoid function and the hyperbolic tangent function respectively while is bias, , and are the weight values used to avoid the gradient loss problem.

c. Prediction Model

Prediction models are built with Keras. Cross-validation is used with multiple data split consist of 80:20 and 70:30. The model is using a sequential model with one conv-LSTM layer. Equations (*1*), (*2*), (*3*), (*4*), (*5*) have already been built inside it. The conv-LSTM layer consisted of 256 filters with a reshape of (1, 1, 1, 1). The shaping is required for convolutional because the input needs to be 5-dimensional. The kernel in the conv-LSTM layer is using 1 width and 1 height to specify the height and width of the convolutional window. The conv-LSTM layer also uses reLU for its activation function for the recurrent step. The first layer is followed by a flattening layer to connect into one dense layer with an 8 unit and reLU activation function. Lastly, there is a one-unit dense layer with Adam optimizer as output layer to predicted blood glucose value. Model is trained with 100 epochs with 128 batch sizes.

d. Flowchart Design

After loading the time series dataset of continuous glucose monitoring (CGM), we need to choose a random patient and set the modelCount to 0. The modelCount is used to choose the best model from the training session for each patient. Whenever the modelCount is lower than 5, we will go to the next step of the training process.

First, we need to construct a training and test set with data split cross-validation depending on the variation that is used right now. There are two cross-validations in this study and both are using data split, it is 70:30 and 80:20 data split. Second, we will construct the conv-LSTM model as described in section 3. Next RMSE will be initialized with a big number and the training process will be started. After the model training is done, the model and the corresponding RMSE will be stored temporally to compare it with the initial RMSE or the previous RMSE model later on. If the new RMSE is greater than the previous one (or initial RMSE if it was the first run), the system will run the training process again. If the new RMSE is lower than the previous one, the current model and RMSE will be stored and the modelCount will be increased by one. After the modelCount is not lower than 5, the system will be outputting the best model for the patient. The process will be repeated until four models of a different patient are acquired for each cross-validation variation. For better understanding, the flow chart for this study algorithm is shown in Figure 2.

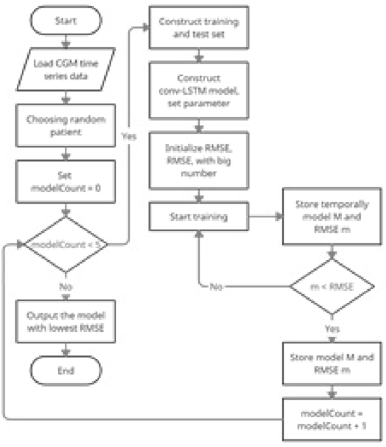

Start
Load CGM time series data
Choosing random patient
Set modelCount = 0
modelCount < 5
Yes
No
Output the model with lowest RMSE
End
Construct training and test set
Construct conv-LSTM model, set parameter
Initialize RMSE, RMSE, with big number
Start training
Store temporally model M and RMSE m
m < RMSE
No
Yes
Store model M and RMSE m
modelCount = modelCount + 1

Figure 2 System Flowchart

e. Datasets

This study used the SENCE public dataset published in 2017 by the JAEB Center for Health Research. The dataset can be accessed in [14]. The data that is used from the SENCE dataset is the CGM. It was a data of glucose per 5 minutes for approximately 6 months to one year from a couple of patients. The data will be processed first into the desired shape as needed for the conv-LSTM layer in the prediction model. Only four patients out of all the patients will be used in this study. The dataset sample is shown in Table *1*.

There are 4 columns in the dataset, RecId is a record id for each data, PtID indicates which patient has data on that row, DeviceDtTm has information about the date and time the data is recorded and the value is the blood glucose that got recorded.

Table 1 Dataset Sample

RecID	PtID	DeviceDtTm	Value
5946141	38	2001-06-19 01:41:08	321.0
5946142	38	2001-06-19 01:46:08	319.0
…	..	..	..
8225193	38	2001-12-21 21:34:46	70.0
82251945	38	2001-12-21 21:39:46	73.0

f. Evaluation Methods

Two evaluation methods will be used in this research. We choose the two most commonly used ones, namely, root means square error and mean absolute error. Even more, root mean square error is usually used in blood glucose prediction research [15].

a. Root Mean Square Error (RMSE)

The variance of all errors rooted by the square is called the RMSE [16]. A given data set is computed by taking the difference between the target and the predicted data to obtain an absolute fit of a model. The relative measure of fit is called root squared, while the absolute measure of fit is called RMSE. It can be used to refer to the standard deviation of unexplained variance.

A better fit is represented by a lower RMSE value. RMSE is an evaluation method to know how good a model is at predicting the data. It's a good indicator of how accurate a model is at handling complex data.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \tag{6}$$

where is the actual data and is the predicted glucose values.

b. Mean Absolute Error (MAE)

The mean absolute errors are known as MAE. It shows the gap between the actual value and the forecasted value. The value of the error from the forecast on average can be expected with MAE.

MAE is also the most popular error measure besides RMSE. MAE is given as:

where is the actual target value for the test instance , while the predictive target value for the test instance is and the number of test instances is denoted by.

$$\boldsymbol{\Sigma}_{xx}(\boldsymbol{\theta}) = \boldsymbol{\Lambda}_x\boldsymbol{\Phi}\boldsymbol{\Lambda}'_x + \boldsymbol{\Theta}_\delta\ , \tag{7}$$

3. Result

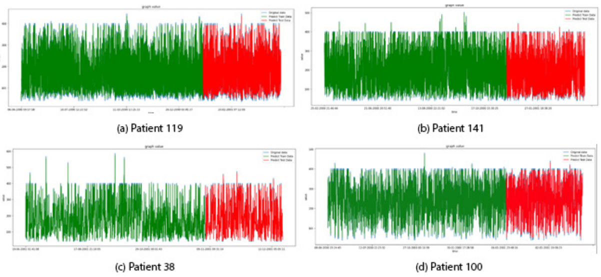
In this section, we explain two cross-validation variations investigated in this study. A model that automatically fits the samples in training data will get an overfitting result. Overfit is a condition in which the model scores a perfect metric but fails to predict anything useful on the data that is not yet visible. Because of that, the use of cross-validation is necessary. We used two different ratios of time-series data split in this study, 70:30 and 80:20.

Many experiments use the two data split ratios of 70:30 and 80:20. The choice often gives pretty good results [5]. Another research conducted by [6] used a data split of 66:34, which is quite close to the data split used in our study.

In Figure 3 and Figure 4, the X-axis represents the timeline (date and time) whereas the Y-axis represents the blood glucose value. Three different data is shown by three different line colors. The blue line indicates the original data from the dataset. The green line is the training data from splitting a portion of the data from the original data. The last line, the red one, is the predicted outcome from the data test

Variation of cross-validation aims to find the best results from the training model. Based on [17] the bigger the training data, the better the model and the predictions, the better the results. However, the biggest size of training data does not mean it has the best results, so we vary the data to find better results.

a. Cross-Validation with Ratio 70:30



(a) Patient 119

(b) Patient 141

(c) Patient 38

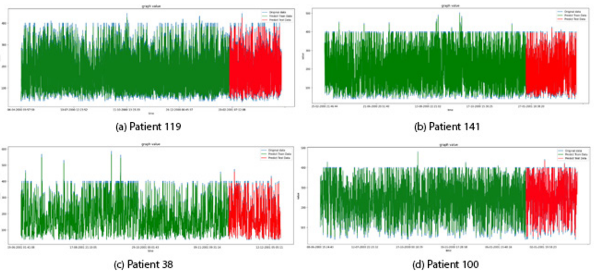
(d) Patient 100

Figure 3 Result graph with 70:30 data split

This section shows the results of the cross-validation for four patients. The instrument that is used to derive results is cross-validation of data split 70:30. The purpose of cross-validation is to acquire more reliable results and to avoid the issue of overfitting.

We can see in *Figure 3* that the prediction result or the red line is almost matched the original data. It means the prediction has a small error.

b. Cross-Validation with Ratio 80:20



(a) Patient 119

(b) Patient 141

(c) Patient 38

(d) Patient 100

Figure 4 Result graph with 80:20 data split

This section shows the results of the cross-validation for the 80:20 data split of the data for four patients. The graph can be seen in *Figure 4*. We can compare them with the other cross-validation result, the prediction result or the red line are not covering up the blue line as much as in *Figure 3*.

From the results of the comparison, it can be seen that the error in the 80:20 cross-validation is greater. These results will be evaluated in the discussion section.

4. Discussion

The data in this study is a time-series data of blood glucose. Time series is well-ordered data so that the kernel size is 1-dimensional convolutional. After reshaping the dataset into 5-dimensional, the algorithm will project the input through the kernel. There is no direct relationship between the kernel size and the result accuracy for the time series because of the 1-dimensional size. Whereas filter parameter affects the accuracy of the model, selecting filter 256 increases the accuracy higher than filters 32, 64, and 128.

Table 2 RMSE and MAE Cross-Validation 70:30

Evaluation Method		Patient ID			
		119	141	38	100
RMSE	Train	21.44	23.99	22.62	21.48
	Test	21.26	23.11	21.58	22.98
MAE	Train	9.44	9.71	8.73	9.02
	Test	9.55	9.53	9.03	9.40

Using the RMSE and MAE evaluation methods from formula (*6*) and (*7*), the results in Table 2 and Table 3 were obtained in the scenario of cross-validation 70:30 and 80:20 respectively.

We can see from Table *2* and Table *3*, calculation for patient 119 gets the best RMSE result from the 70:30 data split because it has the lowest RMSE value. Inversely, in terms of MAE, patient 38, patient 38 gets the best MAE result from the 70:30 split data because it has the lowest MAE value.

Table 3 RMSE and MAE Cross-Validation 80:20

Evaluation Method		Patient ID			
		119	141	38	100
RMSE	Train	21.61	23.94	22.68	21.70
	Test	22.06	23.05	21.97	22.75
MAE	Train	10.32	9.78	9.38	9.00
	Test	10.66	9.52	9.80	9.20

Apart from that, we can see that there is some variance in the individual errors from comparing the RMSE and MAE values of each patient. The greater the difference of the value, the greater the inconsistency of error. The differences in the value of our results are on average 11 to 13. That number is not that big but it is not that small either. So it is not possible to have a very large error in the results.

Overall although sometimes the 80:20 scenario gives better results for some results, the 70:30 scenario gives a more stable good result. Here we can see even though the 80:20 ratio of split data has larger training data sizes, the results are no better than the 70:30 ratio where the size of the training data used is smaller.

The result of RMSE in this study is acceptable when we recall the dependent variable because that is the deciding factor for the RMSE threshold. The RMSE in this study is comparable with the result in [5], but the result from [6] is still has a better result with 12.38 RMSE. The difference in the RMSE between the study is affected by the dataset and the model used.

Comparison with LSTM, conv-LSTM have comparatively good performances in solving time series problems. It also solves the problem of LSTM where it can't access the content of its previous memory cell when its output gate is closed. Although conv-LSTM is usually used for multiple-dimensional data, in the study conv-LSTM was used for one-dimensional cases and still yielded good results.

4. Conclusion

This paper describes the use of a deep neural network as a method of predicting blood glucose. The deep network has a sequential model that consists of one conv-LSTM layer, one flattened layer, one fully connected layer, and one last layer for the output.

The model has successfully attained a good result of RSME and MAE even with two different data split cross-validations. The 70:30 cross-validation gives a more stable result. Since the average range of RMSE and MAE is between 11 to 13, they are not significantly larger or smaller. Thus, the model can minimize errors in the prediction.

The result is affected by problems such as missing data from the patient. Missing data may occur, for example, because the patient takes off the CGM device that records the patient blood glucose.

Reference

[1] WHO NCD Management-Screening, Diagnosis, and Treatment, "Global Report on Diabetes", 2016. Isbn, 9789241565257. Available online at: https://www.who.int/publications/i/item/9789241565257

[2] Q. Wang, S. Harsh, P. Molenaar, and K. Freeman, "Developing personalized empirical models for Type-I diabetes: An extended Kalman filter approach," 2013. American Control Conference, Washington, DC, 2013, pp. 2923-2928, DOI: 10.1109/ACC.2013.6580278.

[3] Mhaskar, Hrushikesh & Pereverzyev, Sergei & Van der Walt, Maria, "A Deep Learning Approach to Diabetic Blood Glucose Prediction," 2017. Frontiers in Applied Mathematics and Statistics. 3. 10.3389/fams.2017.00014.

[4] Novara, Carlo & Pour, Nima & Vincent, Tyrone & Grassi, Giorgio, "A Nonlinear Blind Identification Approach to Modeling of Diabetic Patients" 2015. IEEE Transactions on Control Systems Technology. 19. 1-1. 10.1109/TCST.2015.2462734.

[5] Q. Sun, M. V. Jankovic, L. Bally, and S. G. Mougiakakou, "Predicting Blood Glucose with an LSTM and Bi-LSTM Based Deep Neural Network," 2018. 14th Symposium on Neural Networks and Applications (NEUREL), Belgrade, 2018, pp. 1-5.

[6] T. El. Idriss, A. Idri, I. Abnane, and Z. Bakkoury, "Predicting Blood Glucose using an LSTM Neural Network," 2019. Federated Conference on Computer Science and Information Systems (FedCSIS), Leipzig, Germany, 2019, pp. 35-41.

[7] Hochreiter, Sepp & Schmidhuber, Jürgen, Long Short-term Memory. Neural computation, 1997, 9. 1735-80.

[8] Shi, X.; Chen, Z.; Wang, H.; Yeung, D.; Wong, W.; Woo, W. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In Proceedings of the 28th International Conference on Neural Information Processing Systems; MIT Press: Montreal, QC, Canada, 2015; pp. 802–810.

[9] Rahman, Md. M.; Siddiqui, Fazlul H., "An Optimized Abstractive Text Summarization Model Using Peephole Convolutional LSTM", 2019 Symmetry 11, no. 10: 1290

[10] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[11] Chung, H., & Shin, K. S, "Genetic algorithm-optimized long short-term memory network for stock market prediction," 2018. Sustainability, 10(10), 3765

[12] Goodfellow, I.; Bengio, Y.; Courville, A.; Bengio, Y. Deep Learning; MIT Press: Cambridge, MA, USA, 2016; pp. 373–418

[13] Schmidhuber, J.; Hochreiter, S. Long short-term memory. Neural Comput. 1997, 9, 1735–1780.

[14] Gers, F.A.; Schmidhuber, J.; Cummins, F. Learning to forget: Continual prediction with LSTM. Neural Comput. 1999, 12, 2451–2471.

[12] Kim, Y.; Roh, J.H.; Kim, H. Early Forecasting of Rice Blast Disease Using Long Short-Term Memory Recurrent Neural Networks. Sustainability 2017, 10, 34.

[13] Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W., Woo, W., "Convolutional LSTM network: a machine learning approach for precipitation nowcasting" 2015. Adv. Neural Inf. Process. Syst. 2015 (January), 802–810 Jun

[14] JAEB Center for Health Research. Available online at: https://public.jaeb.org/datasets/diabetes

[15] T. El Idrissi, A. Idri, and Z. Bakkoury, "Systematic map and review of predictive techniques in diabetes self-management", International Journal of Information Management, 2019; vol. 46, pp. 263-277

[16] E. Daskalaki, A. Prountzou, P. Diem, and S. G. Mougiakakou,"Real-Time Adaptive Models for the Personalized Prediction of Glycemic Profile in Type 1 Diabetes Patients," Diabetes Technol. Ther., vol. 14, no. 2, pp. 168–174, 2012.

[17] Medar, Ramesh & Rajpurohit, Vijay & Rashmi, B. (2017). Impact of Training and Testing Data Splits on Accuracy of Time Series Forecasting in Machine Learning. 1-6. 10.1109/ICCUBEA.2017.8463779.

[18] Rahman, Motiur et al. "A deep learning approach based on convolutional LSTM for detecting diabetes." Computational biology and chemistry vol. 88 (2020): 107329. doi:10.1016/j.compbiolchem.2020.107329

Peer Reviewer

The Board of Editors greatly appreciate the participation of the following reviewers that help during the review process for the publication of Khazanah Informatika since 2020.

1. Adi Supriyatna, Universitas Bina Sarana Informatika, Bandung, Indonesia
2. Afandi Nur Aziz Thohari, Politeknik Negeri Semarang, Semarang, Indonesia
3. Akmal Junaidi, Jurusan Ilmu Komputer Universitas Lampung, Indonesia
4. Alwis Nazir, Universitas Islam Negeri Sultan Syarif Kasim Riau
5. Anjar Wanto, STIKOM Tunas Bangsa, Pematangsiantar - Sumatera Utara, Indonesia
6. Ardi Pujiyanta, Universitas Ahmad Dahlan Yogyakarta, Indonesia
7. Aris Rakhmadi, Universitas Muhammadiyah Surakarta, Surakarta, Indonesia
8. Auzi Asfarian, Institut Pertanian Bogor University, Indonesia
9. Bana Handaga, Universitas Muhammadiyah Surakarta, Surakarta, Indonesia
10. Budi Nugroho, Pusat Penelitian Informatika, Badan Riset dan Inovasi Nasional, Jakarta, Indonesia
11. Devi Afriyanti Puspa Putri, Universitas Muhammadiyah Surakarta, Surakarta, Indonesia
12. Diah Priyawati, Universitas Muhammadiyah Surakarta, Surakarta, Indonesia
13. Eka Nila Kencana, Universitas Udayana, Bali, Indonesia
14. Endah Sudarmilah, Universitas Muhammadiyah Surakarta, Surakarta, Indonesia
15. Endang Wahyu Pamungkas, Universitas Muhammadiyah Surakarta, Surakarta
16. Favian Dewanta, Telkom University, Bandung, Indonesia
17. Frieyadie, STMIK Nusa Mandiri Jakarta, Jakarta, Indonesia
18. Gunawan Ariyanto, Universitas Muhammadiyah Surakarta, Surakarta, Indonesia
19. Indra Waspada, Universitas Diponegoro, Semarang, Indonesia
20. Irma Yuliana, Universitas Muhammadiyah Surakarta, Surakarta, Indonesia
21. Iwan Awaludin, Politeknik Negeri Bandung, Bandung, Indonesia
22. Jan Wantoro, Universitas Muhammadiyah Surakarta, Surakarta, Indonesia
23. Lasmedi Afuan, Universitas Jenderal Soedirman, Purwokerto, Indonesia
24. Leon Andretti Abdillah, Bina Darma University, Palembang, Indonesia
25. Leonard Goeirmanto, Universitas Mercu Buana, Jakarta, Indonesia
26. Lutfiyah Dwi Setia, Politeknik Negeri Madiun, Madiun, Indonesia
27. Maryam, Universitas Muhammadiyah Surakarta, Surakarta, Indonesia
28. Mei Silviana Saputri, Universitas Indonesia, Depok, Indonesia
29. Naufal Azmi Verdikha, Universitas Muhammadiyah Kalimantan Timur, Samarinda
30. Nor Bakiah Abd Warif, Universiti Tun Hussien Onn Malaysia, Johor, Malaysia
31. Pristi Sukmasetya, Universitas Muhammadiyah Magelang, Indonesia
32. Ramalia Noratama Putri, Sekolah Tinggi Ilmu Komputer Pelita Indonesia, Pekanbaru, Indonesia
33. Ridho Ananda, Institut Teknologi Telkom Purwokerto, Purwokerto, Indonesia
34. Sayekti Harits Suryawan, Universitas Muhammadiyah Kalimantan Timur, Samarinda, Indonesia
35. Sitaresmi Wahyu Handani, Universitas Amikom Purwokerto, Purwokerto, Indonesia
36. Siti Helmiyah, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
37. Sri Karnila, Institut Informatika Dan Bisnis Darmajaya, Bandar Lampung, Indonesia
38. Sukirman, Universitas Muhammadiyah Surakarta, Surakarta, Indonesia
39. Tati Ernawati, Politeknik TEDC Bandung, Bandung, Indonesia
40. Umi Fadlilah, Universitas Muhammadiyah Surakarta, Surakarta, Indonesia
41. Ventje Jeremias Lewi Engel, Institut Teknologi Harapan Bangsa, Indonesia
42. Wiwit Supriyanti, Politeknik Indonusa Surakarta, Surakarta, Indonesia

43. Yuliant Sibaroni, Telkom University, Bandung, Indonesia
44. Yusuf Sulistyo Nugroho, Universitas Muhammadiyah Surakarta, Surakarta, Indonesia