International Journal of Engineering & Technology, 7 (4.40) (2018) 48-52

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Building TabuCount Application to counting Tabu Search iteration with Borland Delphi 7

Miswanto[1], Bramantara Yudha[2], Sapriyanti[3], Hendra Setiawan[4], Kadinar Novel[5]

[1]STMIK Nusa Mandiri Jakarta
[2] Universitas Indraprasta PGRI Jakarta
[3] STMIK Nusa Mandiri Jakarta
[4] STMIK Bani Saleh Bekasi
[5] AMIK BSI Jakarta
**Corresponding author E-mail:14002109@nusamandiri.ac.id*

Abstract

This research aims to build desktop application called TabuCount with Borland Delphi 7.0 application in the Software Engineering subject. TabuCount helping other researchers or other students who used Tabu Search Algorithm like Travelling Salesman Problem to count each iteration step by step, and minimized human error for counting number. This research used *Research and Development* (R&D) design procedures through 5 steps, which are: (1) data classification, (2) data collecting, (3) design application, (4) field testing, and (5) building application. Based on the results of manual count with Ms. Excel and the TabuCount result, it was obtained the same result are valid and applicable. For an effective based on the research, TabuCount application were effective to use. Therefore, researchers who wanted to use Tabu Search Algorithm (Travel Salesman Problem) is suggested to use TabuCount as an alternative support application.

Keywords: *Borland Delphi, object oriented programming, tabu search iteration, TabuCount application, Travel salesman problem*

1. Introduction

Nowadays, development of the application desktop left behind from web application development, because people called "now era" don't like something application that must be installed. They prefer to access on the internet and browsing for their needed to the internet. For now software based that was installed on a computer or laptop slowly but sure switched to become web application browser. But in terms of programming, desktop applications have a big challenge for programmers to be able developing some applications with their innovations on programming. One of the best software to building some desktop application called Borland Delphi.

In the era 2000 to 2008, we already know that in Software Engineering, there are so many application based on Object Oriented Programming (OOP) like Visual Basic and Borland Delphi. Delphi was originally developed by Borland as a rapid application development tool for Windows based as the successor of Turbo Pascal Application. Borland Delphi implemented full of object-oriented programming to the existing language, and since then the language has grown better to support generics, anonymous methods, and native Component Object Model (COM). In 2006, Borland's developer tools section was transferred from Borland to a wholly owned subsidiary known as Code Gear, then we know which was sold to Embarcadero Technologies in 2008[5].

Borland Delphi is an application Integrated Development Environment (IDE) based program for building an application development of desktop based, web based, mobile based, and console software based. Borland Delphi compilers use their own Object Pascal code of Pascal and generate native code for Microsoft Windows, Mac OS, IOS, Android and Linux. Borland Delphi includes one of best code editor, a visual designer, an integrated debugger, a source code control component, and support for third-party plugins which can compatible in Delphi.

Borland Delphi is an useful application for Building some application to help researcher for their applying some algorithm method from manual to application based on study case. For this research Borland Delphi used to counting Tabu Search iteration between 1 iteration to other iteration in case to find the optimum route that can Engineer take from Head Office to Customer Office like the travel salesman problem. Based on the distance and time problem between Head Office and customer office we assume the optimum route can be describe in this research.

One of algorithms that solves scheduling problem is Tabu Search Algorithm. Tabu Search is a optimization method which is based on local search. The search process starts from one solution to the next solution, by selecting the best neighborhood solution compared to the current solution and is not forbidden solution (Aladag and Hocaoglu, 2007:56)[1].

2. Research Method

Based on the objectives, this research is included in the type of qualitative research tool. This research is descriptive, that is building application with appropriate interpretation with purpose to help the researchers didn't count manually for iteration step in Tabu Search Algorithm. The object of this research is Produce desktop application called TabuCount by using Borland Delphi 7 based Tabu Search Algorithm case study travel salesman problem.

2.1. Field Research

There are so many journal on the internet that show iteration count for Tabu Search Algorithm case study travel salesman problem solving still manual counting, then to minimized human error from calculating value from 1 distance to other distance or time from Head Office to Customer Office, it must be standardized and had a better way for count and calculating that numerous of value.

2.1. Study of Literature

According to Beker (1974)[9], scheduling is an activity to allocate available resource to ensure that the plan can run well with time and energy which are used efficiently. This research used *Research and Development* (R&D) design procedures through 5 steps, which are: (1) data classification, (2) data collecting, (3) design application, (4) field testing, and (5) building an application. Based on the results of manual count with the TabuCount result, it was obtained the same result as valid and applicable. Here’s the diagram of methodology of this research :

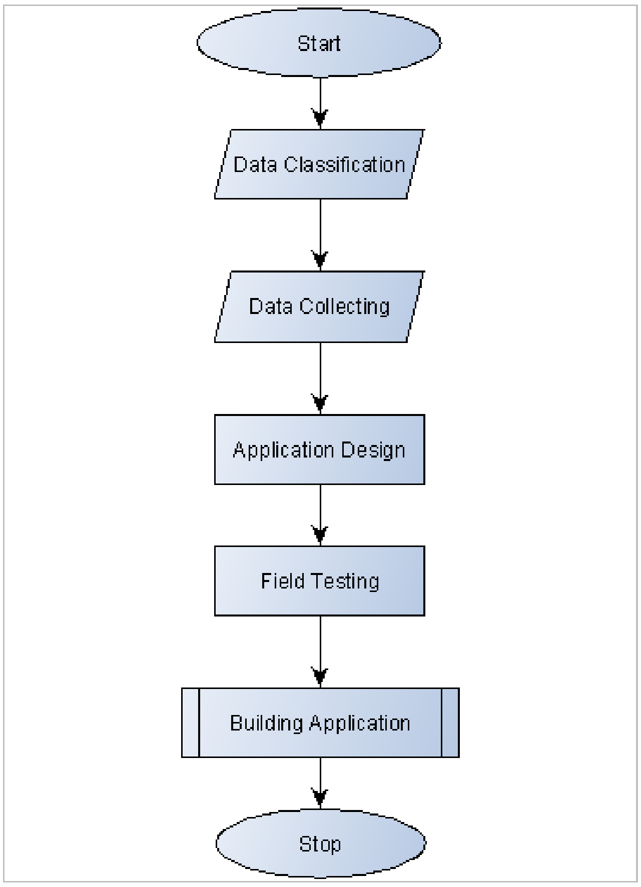


Fig. 1: Flowchart of TabuCount.

3. Research Theory

Borland Delphi version 7 is a very popular desktop application in the era of 2000 to 2008. It was announced in August 6, 2002 which consist of several version such as Delphi 7 Architect, Delphi 7 Enterprise, Delphi 7 Professional and Delphi 7 Personal. Borland Delphi supports Rapid Application Development (RAD) method. Borland Delphi uses the Pascal based programming language called Object Pascal introduced by Borland before. It can supports native cross-compilation and for visual designer has traditionally used Visual Component Library (VCL) for windows.

In order to support development for Microsoft Windows and other system with code developed and other software development tools, Delphi can supports independent interfaces of Component Object Model (COM) too with reference counted class implementations, and also support for many third-party components library. Some interface can be implementation with delegated to fields or from properties of it classes. Message handlers are implemented by tagging a method of a class with the integer constant of the message to handle. Database connectivity is extensively supported through the many VCL database aware and database access components library.

Then for software development, progress maintenance, quality of assurance and cost of poor quality are major groups responsible for overall cost in software production process. The exact proportion among them varies significantly in consecutive software releases, which is caused by many factors, among which are usually: predicted stability and capability of the processes used in developing the product, its size, scope of the changes introduced for the given software new release, and perceived quality of the software new releases which are already used[2].

We can increasing need for the reliability of the software systems status, especially mission critical applications in the public safety domain, raises the better bar for the accuracy of prediction and estimation techniques. They should be evaluated in several dimensions, e.g. the effort needed to perform the tasks, pre and the assumptions, risks defined for the activities, and the quality of both about the product and the processes. The accuracy of estimations process in many areas brings about more concerns regarding techniques already used in the software engineering literature system.

3.1. Delphi Characteristic

Borland Delphi is one of the best high-level programming language, intended to be very easy to use and originally based on Pascal language. Pascal was originally developed as a general purpose language which "suitable for expressing the fundamental constructs known at the time in a concise and logical way", and "its implementation was to be efficient and competitive with existing compilers" but without low-level programming facilities or access to hardware. Pascal and its descendants, including Borland Delphi, can support access to hardware and low-level programming, with the facility to incorporate code written in assembly language and other languages.

Meta classes are the first class objects in Borland Delphi. Some objects are actually references to the other objects, which Delphi implicitly for references, so there is usually no need to manually allocate memory for pointers to objects or use similar techniques that some other languages generally needed. There are some dedicated reference counted string types, and also null terminated strings. Every string can be declared by using plus operator like + sign, better than functions declaration. For dedicated string types Delphi handles memory management without programmer intervention for coding.

An object contains a group of data and methods that acts on the data, a line type might have a function, which is passed point of type Cartesian-coordinate and returns the distance from the line. The resulting code is very highly organized and easy to read and adapt[7]. Delphi includes an integrated IDE function. The Delphi products bundling with a Visual Component Library (VCL), including most of its source code. Third-party components and tools to enhanced the integrated development environment or for other Borland Delphi related development tasks are available. The VCL framework maintains a high level of source compatibility between one versions to another version, which simplifies updating existing or old source code to a newer Borland Delphi version.

Then for third-party libraries maybe need an updates from their vendor, but if source code is supplied, recompilation with the newer version may be supported. The VCL was an early adopter of dependency injection or inversion of control, it uses re-usable component model, extensible by the developer. The compiler was optimizing and single pass. It can optionally compile to a single executable which does not require DLLs file. Delphi itself also can generate standard DLL, ActiveX DLL, COM for server and Windows based services. Borland Delphi IDEs since Delphi 2005 increasingly support refactoring features such as method extraction and the possibility to create UML modeling from the source code or to modify the source through changes made in the model.

3.2. Visual Component Library (VCL)

One of traditionally visual designer used by Delphi is Visual Component Library, VCL is a visual component based object-oriented framework for developing the user interface of Microsoft Windows applications for easy to use[6]. It is written in Object Pascal, the combination of the Borland Delphi language and the VCL framework written in that language addressed these by :

- For streaming framework, allowing an object and sub objects to be streamed to text or binary format – T Component, the root class of the VCL framework.
- For form designer that saved to a stream, saving the description of the objects not the code required to create them, with objects capable of self-creation when streaming the description back in.
- Then language features that allowed instances of a class to be customized without sub classing, in the two ways: first, through the use of properties, allowing instances of an object to have its own fields easily changed through streaming; second, by allowing events method pointers called at specific places in code which can handle by other object; then the method show pointers were attached to become another instance. In other words, this creates "custom actions through delegation instead of inheritance".
- Then controls that wrapped native Windows controls, in framework that managed windows resources.

3.3. Tabu Search Algorithm

Tabu Search was introduced for the first time by Glover in 1986, tabu Search is an algorithm which is in scope of heuristic method. The basic concept of Tabu Search is an algorithm which guides its stages in order to create the most optimum aim function without being stuck in initial solution which is discovered while the stage is taking place. The aim of this algorithm is to prevent repetition and to find similar solution in an iteration which will be used again in the next iteration. The word Tabu or Taboo comes from Tongan language, a Polynesian language which is used by Aborigine tribe of Tonga island to indicate something that shall not be “touched” due to its sanctity (Glover and Laguna, 1997)[8]. According to Webster dictionary, tabu means prohibition that is forced by a social culture as a prevention or something which is forbidden because it is dangerous. The danger that must be avoided in Tabu Search is inappropriate trip route and being stuck without escape.

The basic form of Tabu Search (TS) is founded on ideas proposed by Fred Glover (1989)[3]. The method Tabu use based on procedures was designed to cross boundaries of feasibility or local optimality, instead of treating them as barriers. For an examples of these procedures, derived from surrogate constraint method and cutting plane approaches, systematically impose and released constraints to permit exploration of otherwise forbidden regions. TS is based on the premise that problem solving like Travel Salesman Problem, in order to qualify as intelligent, must incorporate adaptive memory and responsive exploration.

Allocation memory features of Tabu Search allows the implementation of procedures that are possible of searching the solution space effectively and efficiently. Since local choices are guided by information collected during the search, Tabu Search contrasts with memory less designs that heavily rely on semi random processes that implement a form of sampling. The repression based on responsive exploration in Tabu Search, which is in a deterministic or probabilistic implementation, derives from the supposition that a bad strategic choice can often yield more information than a good random choice, and therefore provides a basis for progressively improved strategies that take advantage of search history. A Tabu Search Algorithm (TSA) is a local search algorithm that allows non improving moves when a local optimum is encountered, in the hope that in this way the solution will be improved[10].

The main idea of Tabu Search Algorithm (TSA) is as follows: starting from an initial solution, choosing a series of specific search direction (or called movement) as a temptation, TSA chooses to implement the movement which can reduce the objective function value the most. To avoid to fall into the best local solution, TSA adopts a flexible "memory" technology, which records and selects the optimized process, so as to guide the direction of the next search, which is the establishment of the tabu list. Tabu list limits a movement’s availability to be utilized for a period of time. A movement, so long as it is in the tabu list, cannot be implemented in the current iteration, so that the revisit of algorithm can be avoided since it has visited the solution during several recent iterations, which helps algorithm to be prevented from converging to local optimal solution.

However, TSA has stronger dependence on the initial solution. A good initial solution can help TSA to find a good solution in the solution space, while poor initial solution can reduce the convergence rate of TSA. Normally, while solving some particular problem, it better to use other optimization algorithms to generate an first solution of high quality measurement. Beginning with first solution before, Tabu Search can improve the quality and efficiency of local searching. There is no doubt that the greed of TSA will rob the algorithm of global optimization capability [4]. In the best way, Tabu Search can be viewed as beginning in the same way as an ordinary local or neighborhood search, proceeding iteratively from one point (solution) to another point until a chosen termination criterion is satisfied. Each one solution x has an associated neighborhood $N(x) \subset X$, and each other solution $x' \in N(x)$ is reached from x by an operation called a *move.*

4. Analysis and Design

This research supposed to build an application that can make other researcher easy on calculating Tabu Search iteration using this software. This software build with object of 1 head office with 5 customer office who wanted to be maintenance using engineers scheduling.

4.1 Identification Problems

There are some problem faced by an author when collecting and classification data using Tabu Search Algorith method**.** The problem generally are :

- There’s no application to counting iteration, the value of iteration itself consist of various data.
- For counting iteration still using manually method like Microsoft Excel calculation, and it needed something new and simple.
- Minimalize human error when assigning value of data for research.

4.2 Research and Development

This research used Research and Development (R&D) design procedures through 5 steps, which are: (1) data classification, (2) data collecting, (3) design application, (4) field testing, and (5) building application. The step by step for building TabuCount are :

4.2.1 Data Classification, this research classify between distance and time allocation from 1 Head Quarter Office to 5 customer office which get support maintenance at the time.

4.2.2 Data Collecting, this research collecting how far distance and time from 1 Head Quarter to 5 customer office which get support maintenance.

4.2.3 Design Application, this research design an application to counting value about distance and time with simple way to minimalize human error for count number. Distance every node 1 by 1 to another node and give the result at the end.

4.2.4 Field Testing, this research get the white box and black box testing about application function and make sure the application applicable to use.

4.2.5 Building Application, this research had the output application can be use for counting Tabu Search Algorithm iteration with distance and time variable. For the other research who use TSA for their research must be try this simple application**.**

4.3 Design Application

This research will make sure the distance and the time allocation from Head Quarter Office to Customer Office must be set. It can set for the programming issue for TabuCount application. For the distance had already set by this research are :

Table 1. Data of distance

CS	***d(0)***	***d(1)***	***d(2)***	***d(3)***	***d(4)***	***d(5)***
(0)	0	3.5	6.5	8.7	2.9	5.9
(1)	3.5	0	2.7	10	4.7	9.2
(2)	6.5	2.7	0	12.3	9.6	12
(3)	8.7	10	12.3	0	8.5	13.2
(4)	2.9	4.7	9.6	8.5	0	5
(5)	5.9	9.2	12	13.2	5	0

CS for Customer, d(0) for first distance, d(1) for second distance, d(2) for third distance, d(3) for fourth distance, d(4) for fifth distance and the sixth distance is d(5).

Table 2. Data of trip time

CS	***t(0)***	***t(1)***	***t(2)***	***t(3)***	***t(4)***	***t(5)***
(0)	0	5.5	9.5	8.7	4.3	8.4
(1)	5.5	0	4.1	10	7.1	13.8
(2)	9.5	4.1	0	18,5	14.4	18
(3)	13.1	15	18.5	0	12.8	19.8
(4)	4.3	7.1	14.4	8.5	0	7.5
(5)	8.4	13.8	18	13.2	7.5	0

CS for Customer, t(0) for first trip time, d(1) for second trip time, d(2) for third trip time, d(3) for fourth trip time, d(4) for fifth trip time and the sixth trip time is d(5). For first step from this research was to define the initial route / first route and set is as best solution. The initial route was defined by using nearest neighborhood and it was found that initial route was 0 – 4 – 1 – 2 – 3 – 5 – 0 and for that solutions we have the first value for :

Distance (d): 2,9 + 4,7 + 2,7 + 12,3 + 13,2 + 5,9 = **41,7 km**
Trip time (t): 4,3 + 7,1 + 4,1 + 18,5 + 13,2 + 8,4 = **55,6 minutes.**

For the step counting this application running from CS (0) – (5) left to the right d(0) – d(5) and t(0) – t(5) with value of each columns and row consist of. So if we wanted to get the optimum value of distance and trip time, we must make several iteration to move 1 node to another node. Such as in the beginning we change position number 4 and 1 from the Initial Route (0-4-1-2-3-5-0) with (0-1-4-2-3-5-0) or change position number 4 and 2 from the initial route (0-4-1-2-3-5-0) with (0-2-1-4-3-5-0)

And for searching neighbor local distance and trip time, this research trying to make 40 iteration with change position until 40 iteration was shown the optimum calculation of Tabu Search Algorithm. Check Table 3 for the details.

Table 3. Switching for Local Searching Tabu

Switching	Trip Route	d(Km)	t(Minute)
1. Switch 4, 1	0-1-4-2-3-5-0	49,2	67,1
2. Switch 4, 2	0-2-1-4-3-5-0	41,5	55,1
3. Switch 4, 3	0-3-1-2-4-5-0	41,9	57,5
4. Switch 4, 5	0-5-1-2-3-4-0	41,5	57,6
5. Switch 1, 2	0-4-2-1-3-5-0	44,3	59,4
6. Switch 1, 3	0-4-3-2-1-5-0	41,5	61,9
7. Switch 1, 5	0-4-5-2-3-1-0	45,7	63,8
8. Switch 2, 3	0-4-1-3-2-5-0	47,8	71,3
9. Switch 2, 5	0-4-1-5-3-2-0	48.8	73
10. Switch 3, 5	0-4-1-2-5-3-0	44,2	62

After we get the pattern for switching 1 node to another node, so the other step we can build 7 field which can be calculated one to another and there's 2 field show the result of distance and time of trip. Then there's 9 edit text for input/result count, 5 button for Count, Reset, Exit, Clear Iteration List and Input Distance and Time button on the applications and 1 list box for show how much iteration already count. Here's the design of TabuCount Application running form if we running the application from Executable Program.

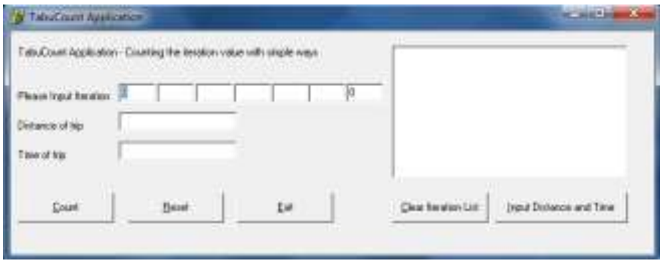
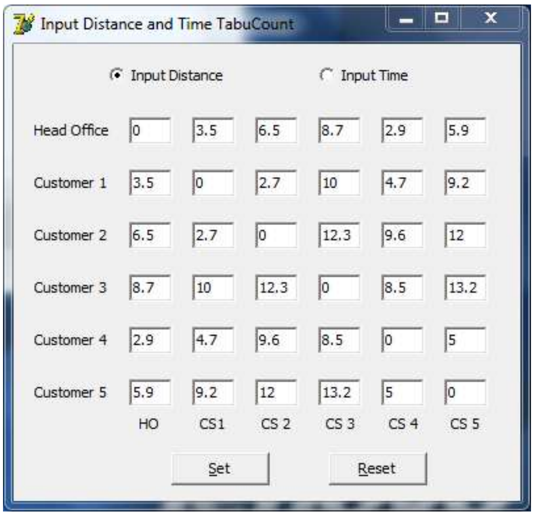


Fig. 2. Design of TabuCount

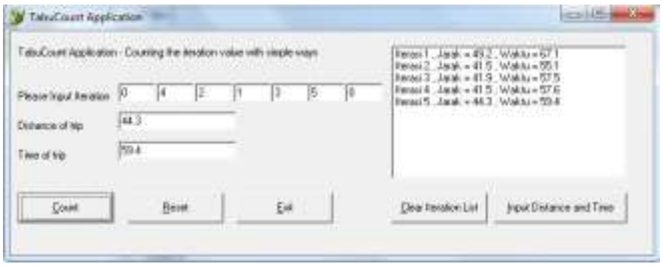
After form design was made in Borland Delphi application, then we must set every field with value same as shown on Table 1 and Table 2 into TabuCount application, for input value of the distance and time we created Form for Input Distance and Time in current TabuCount application. Input for this application was made into another Borland Delphi form which can be initiated as we can set manually or we can change the value for it's default. Here's the design Input Distance and Time form shown :



	HO	CS1	CS 2	CS 3	CS 4	CS 5
Head Office	0	3.5	6.5	8.7	2.9	5.9
Customer 1	3.5	0	2.7	10	4.7	9.2
Customer 2	6.5	2.7	0	12.3	9.6	12
Customer 3	8.7	10	12.3	0	8.5	13.2
Customer 4	2.9	4.7	9.6	8.5	0	5
Customer 5	5.9	9.2	12	13.2	5	0

Fig. 3. Input Distance and Time for TabuCount

After Tabu Search Procedures already initially value of each distance and time then do the field testing every switching and iteration and compared with manual counting like Ms Excel. After several testing and commissioning, the result from manual counting and TabuCount Application was match and had the same value. Here's the testing result for the output of TabuCount application and manual counting from Ms Excel :



TabuCount Application - Counting the iteration value with simple ways

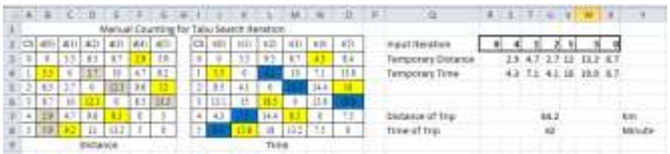
Please Input Iteration: 0 | 4 | 2 | 1 | 3 | 5 | 0

Distance of trip: 44.3

Time of trip: 59.4

```
Iterasi 1 , Jarak = 49.2 , Waktu = 67.1
Iterasi 2 , Jarak = 41.5 , Waktu = 55.1
Iterasi 3 , Jarak = 41.9 , Waktu = 57.5
Iterasi 4 , Jarak = 41.5 , Waktu = 57.6
Iterasi 5 , Jarak = 44.3 , Waktu = 59.4
```

Fig. 4. Testing Result 1 with TabuCount



Manual Counting for Tabu Search Iteration

C	d(0)	d(1)	d(2)	d(3)	d(4)	d(5)
0	0	3.3	6.5	8.7	2.9	5.9
1	3.3	0	2.7	10	4.7	8.2
2	6.5	2.7	0	12.3	9.6	12
3	8.7	10	12.3	0	8.5	13.2
4	2.9	4.7	9.6	8.5	0	5
5	5.9	8.2	12	13.2	5	0

Distance

C	t(0)	t(1)	t(2)	t(3)	t(4)	t(5)
0	0	3.3	9.3	8.7	4.3	8.4
1	3.3	0	[illegible]	15	7.1	11.8
2	9.3	4.1	0	[illegible]	14.4	18
3	13.1	15	18.5	0	12.8	[illegible]
4	4.3	[illegible]	14.4	8.5	0	7.5
5	[illegible]	11.8	18	13.2	7.5	0

Time

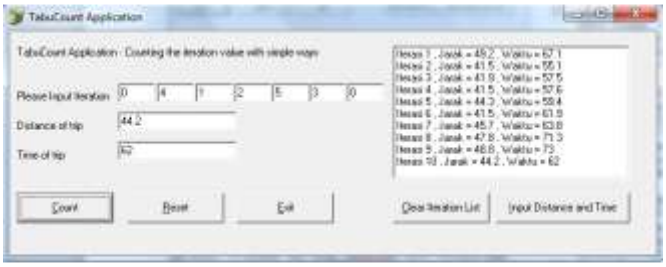
Input Iteration	0	4	1	2	5	3	0
Temporary Distance		2.9	4.7	2.7	12	13.2	8.7
Temporary Time		4.3	7.1	4.1	18	19.8	8.7

Distance of trip 44.2 Km

Time of trip 62 Minute

Fig. 5. Testing Result Manual with Ms Excel

As we can see from Table 3, from result in the Table match with TabuCount iteration, and if we compared with manual counting Ms Excel, we assume that using TabuCount more effective than using manual counting like Ms Excel.



TabuCount Application - Counting the iteration value with simple ways

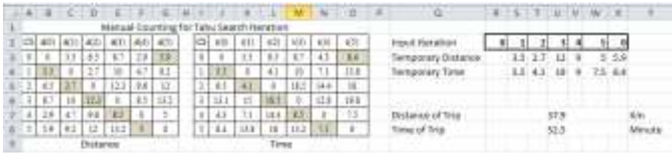
Please Input Iteration: 0 | 4 | 1 | 2 | 5 | 3 | 0

Distance of trip: 44.2

Time of trip: 62

```
Iterasi 1 , Jarak = 49.2 , Waktu = 67.1
Iterasi 2 , Jarak = 41.5 , Waktu = 55.1
Iterasi 3 , Jarak = 41.9 , Waktu = 57.5
Iterasi 4 , Jarak = 41.5 , Waktu = 57.6
Iterasi 5 , Jarak = 44.3 , Waktu = 59.4
Iterasi 6 , Jarak = 41.5 , Waktu = 61.9
Iterasi 7 , Jarak = 45.7 , Waktu = 63.8
Iterasi 8 , Jarak = 47.8 , Waktu = 71.3
Iterasi 9 , Jarak = 48.8 , Waktu = 73
Iterasi 10 , Jarak = 44.2 , Waktu = 62
```

Fig. 6. Testing Result 2 with TabuCount



Manual Counting for Tabu Search Iteration

C	d(0)	d(1)	d(2)	d(3)	d(4)	d(5)
0	0	3.3	6.5	6.7	2.9	5.9
1	3.3	0	2.7	10	4.7	8.2
2	6.5	2.7	0	12.3	9.6	12
3	6.7	10	12.3	0	8.5	13.2
4	2.9	4.7	9.6	8.3	0	5
5	5.9	8.2	12	13.2	5	0

Distance

C	t(0)	t(1)	t(2)	t(3)	t(4)	t(5)
0	0	3.3	8.7	6.7	4.3	8.4
1	3.3	0	4.1	15	7.1	11.8
2	8.1	4.1	0	18.5	14.4	18
3	13.1	15	18.5	0	12.8	19.8
4	4.3	7.1	14.4	8.5	0	7.5
5	8.4	11.8	18	19.2	7.5	0

Time

Input Iteration	0	1	2	1	4	5	0
Temporary Distance		3.3	2.7	12	9	5	5.9
Temporary Time		3.3	4.1	18	9	7.5	8.4

Distance of Trip 37.9 Km

Time of Trip 52.3 Minute

Fig. 7. Testing Result Manual with Ms Excel

After several testing of Count then testing for function of application and make sure the application working fine and had the best result. Then TabuCount.exe was build for everyone who needed it for their research about Tabu Search Algorithm.

5. Conclusion

The TabuCount Application has proved to be very useful tool in calculating measurement of Tabu Search Iteration with 2 variable distance and time being set manually like Travelling Salesman Problem study case. From this research, TabuCount Application maybe give an idea to other researcher to building some desktop application who can be used for other researcher. Therefore, researchers who wanted to use Tabu Search Algorithm (Travel Salesman Problem study case) is suggested to use TabuCount as an alternative support application for their research tools and we hope this software can be helpful. We need feedback from other researcher for the better application can be used for supporting tool for research.

Acknowledgement

This research was collaborated with students Postgraduate of STMIK Nusa Mandiri, Jakarta class of 14.2B.01. Thank you for our Postgraduate lecturer in STMIK Nusa Mandiri Jakarta for your best support for us.

References

[1] Aladag C.H. and Hocaoglu G,"A Tabu Search Algorithm to Solve Course Timetabling Problem", *Hacettepe Journal of Mathematics and Statistics*, Vol.36, (2007), pp.53-64.

[2] M G Stochel, "Reliability and accuracy of the estimation process", *35th IEEE Annual Computer Software and Application Conference*, Vol.35, (2011), pp.350-359.

[3] Glover. F, "Tabu Search, Part I", *ORSA Journal on Computing* 1, Vol. 1, (1989), pp:190-206.

[4] Xinmeng Zhu, Yongsheng Ding, Xin Cai, Haishan Wang and Xiangfei Zhang, "Optimal Schedule for Agricultur Machinery using an improved Immune-Tabu Search," *Proceedings of 36th Chinese Control Conference, Dalian, China*, (2017), pp. 2824 – 2829.

[5] https://en.wikipedia.org/wiki/Delphi_(IDE) last visit : August, 22nd 2018.

[6] https://en.wikipedia.org/wiki/Visual_Component_Library last visit : August, 22nd 2018.

[7] Paul F McCombie and Jim Penman, "The production of interactive engineering design software using Borland Delphi", *ELSEVIER Advanced in Engineering Software*, Vol.32, (2001), pp.789-796.

[8] Glover, F and Laguna, M. "*Tabu Search"*, Kluwer Academic Publisher, (1997).

[9] Beker, Kenneth R, "Introduction to Sequencing and Schedulling", John Wiley & Sons Inc, (1974).

[10] Jelmer P. van der Gaast,Cornelius A. Rietveld, Adriana F. Gabor, and Yingqian Zhang, "A Tabu Search Algorithm for application placement in computer clustering", *ELSEVIER Computer & Operations Research*,Vol. 50, (2014), pp.38-46.