

# **RESEARCH PROPOSAL**

**Doctoral Program in Computer Science**



**A Research Proposal Template for CS Students at BINUS**

**Benfano Soewito 0123456789**

Promotor

Prof. Bahtiar Saleh Abbas, PhD 0325115603

Dr. L. Gaol, Ford 0300000000

Universitas Bina Nusantara

October 2014

A RESEARCH PROPOSAL TEMPLATE FOR CS STUDENTS AT BINUS

A PROPOSAL RESEARCH  
SUBMITTED TO THE DEPARTMENT OF DOCTORAL PROGRAM  
IN COMPUTER SCIENCE  
AND THE COMMITTEE ON GRADUATE PROGRAM  
OF BINA NUSANTARA UNIVERSITY  
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE ADMISSION TO THE DEGREE OF  
DOCTOR OF PHILOSOPHY WHICHEND

Benfano Soewito

Date:

Prof. Bahtiar Saleh Abbas, PhD  
0325115603

Dr. L. Gaol, Ford  
030000000

# A Research Proposal Template for CS Students at BINUS

Benfano Soewito  
Possible thesis advisor: Gaol L. Ford

November 18, 2014

## 1 Introduction

Creation of a formal research proposal is a daunting task; while authors such as Zobel [12] discuss the basic mechanics of technical writing, and writers such as Griffith et al. [6] explain the basics of using L<sup>A</sup>T<sub>E</sub>X for such writing, there is little guidance on the actual structure of a research proposal. This structure may vary considerably depending on the nature of the research topic—whether it consists of a coding project, a theoretical study, a literature survey or other acceptable topics. Despite their differences, all proposals share common features, and this paper provides a L<sup>A</sup>T<sub>E</sub>X template suitable for most such projects. It is expected that a user will modify this template according to the specifics of the proposed research; in particular, boiler-plate section headings and subheadings should be replaced by informative ones specific to the topic.

### 1.1 Problem statement

The introduction of a proposal typically acquaints the reader with the nature of problem being addressed and the basic idea of the project being proposed to address it. It should hit many of the points of the abstract while giving more of the motivation and need for the work being proposed. It should then lay out the structure of the paper, telling the reader what will be found in the sections which follow. Thus, the next sections should address the following topics:

- the thesis statement and research objectives;
- prior work and current art | other approaches to the problem and their drawbacks;

- methodology and experiment—including background research to be done, necessary equipment and algorithmic techniques, and general plan-of-attack;
- analysis and validation—how results will be judged and interpreted;
- a time-table for achieving key objectives, and one or more fallback plans in cases where goals are not met by specific target dates;
- concluding remarks.

## 1.2 Variable Observed

What you are trying to explain and why, as well as some sense of the range of variation in the dependent variable. Remember you are trying to convince others that this is a question worth asking (and answering).

## 1.3 Implications

The proposal should also include a bibliography/reading-list. In this case, it is permissible to include references which are not directly cited in the text. (You can use the `\nocite{*}` command before the bibliography section, as below, to include all BibTeX references from a database file in the bibliography.) You should still cite references in the text as appropriate.

# 2 Prior Work

Depending on the nature of the topic, prior work should either precede or follow the thesis statement. Some topics will require the background information to put the thesis proposal in context. Others are best served by giving the thesis statement first and then contrasting and comparing it with prior work in the field.

## 2.1 T<sub>E</sub>X

If this were a real proposal dealing with how to write proposals, it might be appropriate to say a few words about Knuth, inventor of the T<sub>E</sub>X typesetting language [8]. Description of Knuth's contribution goes here.

## 2.2 L<sup>A</sup>T<sub>E</sub>X (Lamport, 1984)

Description of Lamport's contribution [9] goes here.

## 2.3 Other Work

Description of other contributions leading up to this thesis goes here, e.g., Zobel's book on writing for computer science [12].

# 3 Methodology

It is difficult for students to begin a proposal without an adequate guide; my project, therefore, will demonstrate the following thesis:

It is both simple and useful to provide students with a L<sup>A</sup>T<sub>E</sub>X template for their formal thesis proposals. This will result in far fewer questions from students uncertain as to what belongs in their proposals. Such a template will be of benefit to students in Computer Science/Applied Computing as well as to any others willing and able to use L<sup>A</sup>T<sub>E</sub>X for their work.

A formal thesis statement should be a *falsifiable* statement about the goal you will attempt to achieve with your research project. For a purely scientific project, this is the hypothesis you are testing with your research. For an applied programming project, it is usually a statement about the feasibility and correctness of your approach and the advantages it has over other approaches. For a survey or study, it is usually a statement regarding the need or usefulness of such a study, its intended audience, and so on.

Often, you may want to include an itemized list of goals you plan to achieve. Thus, this paper has the goals of

- helping students, by ridding their minds of confusion;
- helping faculty, too, by eliminating their need to say \No, no, no! You need to include a section on bleh and blah! I told everybody that. I thought I told everybody that.

Arrgghh!!!!"



Figure 1: Include images in a proposal as appropriate

In summary, this section gives the main idea of what you propose, with the goals and contributions it will make to the eld; the details of proposed implementation and methodology will be given later in Sec. 4.

If you didn't already discuss prior work, this is a good place to do so.

## 4 Your Proposed Explanation

Here, you should lay out the details of how you propose to solve the problem and otherwise conduct the research necessary to support your thesis. Include details regarding hardware and software you will use, resources you will draw on, algorithms you will implement, and other ideas about how you will accomplish your task. It is inevitable that your final work will deviate from earlier plans, as you research your topic, learn new methods, and discover what works as expected.

### 4.1 Using Tables and Figures

Use tables and gures as appropriate; a picture can explain a lot in very compact form, and can keep readers interested. Avoid things that are merely ash and do not add any relevant information. (See Fig. 1.)

```

#define D define
#D Y return
#D R for
#D e while
#D I p r i n t f
#D l int
#D W i f
#D C y=v+111;H(x,v) y++= x
#D H(a,b)R( a=b+11;a<b+89;a++)
#D s ( a ) t=scanf ( "%d" ,&a )
#D U Z I
#D Z I ( " 123\
45678\n");H(x,V){ putchar ( ".XO"[*x]); W((x-V)%10==8){ x +=2; I("%d\n" ,(x-V)/10 -1);}}
l V [1600] ,u,r[]={ -1 , -11 , -10 , -9 ,1 ,11 ,10 ,9} ,h []={11 ,18 ,81 ,88} , ih []={22 ,27 ,72 ,77} ,
bz ,lv =60 ,*x ,*y,m,t;S(d,v,f,_,a,b)l*v;{l c=0 ,*n=v+100 ,j=d<u -1?a: -9000 ,w,z,i,g,q=
3-f;W(d>u){R(w=i =0;i <4; i ++ ) w +=( m=v[h[i ]])= f ?300: m==q ? -300:( t=v[ih[i ]])= f? -50:
t==q ?50:0; Y w;}H(z ,0){ W(E(v,z,f ,100)){ c ++;w= -S(d+1,n,q,0,-b,-j);W(w>j){g=bz=z;
j=w;W(w >=b | | w >=8003) Y w ;}}} W(!c){g =0; W(_){H(x,v)c+= *x==f ?1:* x==3 -f ? -1:0;Y c >0?
8000+ c:c -8000;} C;j= -S(d+1,n,q,1,-b,-j )} bz=g;Y d >=u -1? j+(c < <3);j;} main 0{ R;t<
1600; t +=100) R(m=0;m <100; m++) V[t+m]=m <11 | | m >88 | |( m +1)%10 <2?3:0; I(" Level : ");V
[44]
=V [55]=1; V [45]= V [54]=2; s(u);e(lv >0){ Z do{I("You : ");s(m );} e(!E(V,m ,2 ,0)&& m !=99);
W(m !=99) lv --;W(lv <15&&u <10) u +=2; U("Wait\n");I("Value:%d\n",S(0,V ,1 ,0 , -9000 ,9000
)); I("move : %d\n" ,(lv -=E(V,bz ,1 ,0) , bz ));}} E(v,z,f,o)l*v;{l*j,q=3-f,g=0,i,w ,*k=v
+z;W(*k ==0) R(i =7;i >=0;i - -){j=k+(w=r[i ]); e(*j==q)j+=w;W(*j==f&&j-w!=k){W(!g){g=1
;C;}e(j!=k )*(j -w)+o)=f ;}} Y g;}

```

Figure 2: This mystery code ( c 1987 Roemer B. Lievaart) was included from a source file. The L<sup>A</sup>T<sub>E</sub>X file also shows how to change the font-size for a code-listing.

## 4.2 Working with Code and Pseudocode

If you have source code to include, you can do so using the listings package, which will format short inline code fragments such as

```

for ( int i = 0; i < n ; i++)
    cout << "It's easy to add source code to LaTeX documents" ;

```

or simply using the verbatim environment, which gives a Courier font to literal text:

```

for (int i = 0; i < n; i++)
    cout << "It's easy to add source code to LaTeX documents";

```

The listing environment is good for longer code examples and for use in figures, such as in Fig. 2.

Another frequent need is to include algorithms written in pseudocode. The algo package can be used to format algorithms presentably in your documents: Fig. 3 on Pg. 8 shows an example of this. For more options on these packages, consult online resources.

**Algorithm** *CalculateMutationAdequacy*( $T; P; M_o$ )

( Calculation of Strong Mutation Adequacy )

**Input:** Test Suite  $T$ ;

Program Under Test  $P$ ;

Set of Mutation Operators;  $M_o$

**Output:** Mutation Adequacy Score;  $MS(P; T; M_o)$

```

1.  $D \leftarrow \mathbb{Z}n \times s$ 
2.  $\varepsilon \leftarrow \mathbb{Z}s$ 
3. for  $l \in \text{ComputeMutationLocations}(P)$ 
4.   do  $\Phi_p \leftarrow \text{GenerateMutants}(l, P, M_o)$ 
5.   for  $\phi_r \in \Phi_p$ 
6.     do for  $T_f \in \langle T_1, \dots, T_s \rangle$ 
7.       do  $R_f^P \leftarrow \text{ExecuteTest}(T_f, P)$ 
8.        $R_f^{\phi_r} \leftarrow \text{ExecuteTest}(T_f, \phi_r)$ 
9.       if  $R_f^P \neq R_f^{\phi_r}$ 
10.        do  $D[f][r] \leftarrow 1$ 
11.        else if  $\text{IsEquivalentMutant}(P, \phi_r)$ 
12.          do  $\varepsilon[r] \leftarrow 1$ 
13.  $D_{\text{num}} \leftarrow \sum_{r=1}^s \text{pos}(\sum_{f=1}^n D[f][r])$ 
14.  $D_{\text{num}} \leftarrow \sum_{r=1}^s \varepsilon[r]$ 
15.  $MS(P, T, M_o) \leftarrow \frac{D_{\text{num}}}{(|\Phi_P| - \varepsilon_{\text{num}})}$ 
16. return  $MS(P, T, M_o)$ 

```

Figure 3: Algorithm for the Computation of Mutation Adequacy. This example pseudocode is by Gregory M. Kapfhammer.

## 5 Research Timetable

Give a brief overview of how you will proceed to accomplish the project, including a rough schedule for accomplishing the following tasks:

- background research,
- final proposal completion,
- proposal defense,
- research intermediate and final goals, and
- writing of key chapters, leading to final written thesis.

The timetable should take into account the actual schedule followed by the department, CS600 in the fall is devoted to the background research, final proposal and its defense,



beginnings of primary project work and writing of first two chapters. CS601 in the spring is spent finishing the research and the writing and preparing for the thesis defense.

## **6 About Student**

### **6.1 Education**

Master in Computer at Universitas Indonesia	Jakarta	Computer Science	2010
Bachelor in Math at Institut Teknologi Bandung	Bandung	MIPA	2007

### **6.2 Working Experience**

### **6.3 Project/research**

### **6.4 List of Publication**

## References

- [1] H. Blum. A transformation for extracting new descriptors of shape. In W. Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Forms*, pages 362{380. MIT Press, Amsterdam, 1967.
- [2] C. A. Burbeck, S. M. Pizer, B. S. Morse, D. Ariely, G. S. Zauberman, and J. Rolland. Linking object boundaries at scale: a common mechanism for size and shape judgments. Technical Report TR94-041, Department of Computer Science, University of North Carolina - Chapel Hill, 1994. Available online at <ftp://ftp.cs.unc.edu/pub/publications/techreports/94-041.ps.Z> (accessed 12 February 2006).
- [3] James N. Damon. Properties of ridges and cores for two-dimensional images. Technical report, Department of Mathematics, University of North Carolina at Chapel Hill, 1995.
- [4] D. Fritsch, E. Chaney, A. Boxwala, M. McAuli e, S. Raghavan, A. Thall, and J. Earnhart. Core-based portal image registration for automatic radiotherapy treatment verification. *International Journal of Radiation, Oncology, Biology, Physics; special issue on Conformal Therapy*, 5(33):1287{1300, 1995.
- [5] Jacob Furst. *Height Ridges of Oriented Medialness*. Ph. D. Thesis, University of North Carolina, Dept. of Computer Science, 1999.
- [6] David F. Griffiths and Desmond J. Higham. *Learning L<sup>A</sup>T<sub>E</sub>X*. SIAM, Philadelphia, 1997.
- [7] Robert Grzeszczuk, Chris Henn, and Roni Yagel. *Advanced Geometric Techniques for Ray Casting Volumes*. SIGGRAPH 98 Course Notes #4, ACM, July 1998.
- [8] Donald E. Knuth. *The T<sub>E</sub>Xbook*. Addison-Wesley, Reading, Mass., 1984.
- [9] Leslie Lamport. *L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System (2nd Edition)*. Addison-Wesley, Reading, Mass., 1994.
- [10] M. Lipschutz. *Differential Geometry*. Shaum's Outline Series. McGraw-Hill, New York, 1969.
- [11] Plato. *The Republic*, volume VII. Athens, Greece, 360 B.C.
- [12] Justin Zobel. *Writing for Computer Science*. Springer-Verlag, 1997.