



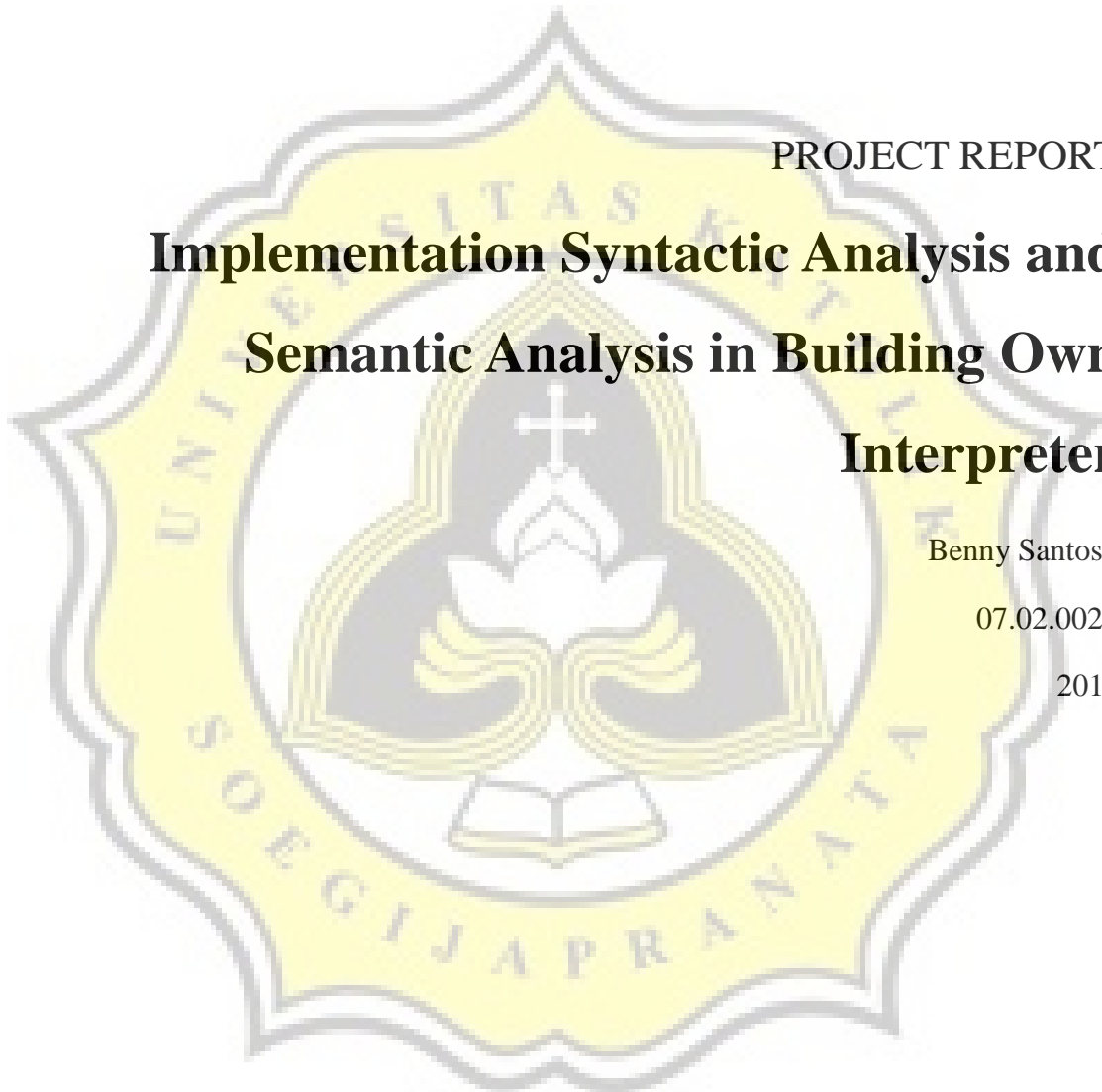
PROJECT REPORT

**Implementation Syntactic Analysis and  
Semantic Analysis in Building Own  
Interpreter**

Benny Santoso

07.02.0020

2011



**FACULTY OF COMPUTER SCIENCE**

**SOEGIJAPRANATA CATHOLIC UNIVERSITY**

Jl. Pawiyatan Luhur IV/1, Bendan Duwur, SEMARANG 50234

Telp. 024-8441555 (hunting) Web: <http://www.unika.ac.id>

Email: [ikom@unika.ac.id](mailto:ikom@unika.ac.id)

# APPROVAL AND RATIFICATION PAGE

## PROJECT REPORT

### Implementation Syntactic Analysis and

### Semantic Analysis in Building Own

### Interpreter

This project report has been approved and ratified by the Dean of faculty of Computer Science and Supervisor on January, 20<sup>th</sup> 2011

With Approval,

Examiner

Examiner

Suyanto EA.,Ir.M.Sc

NPP : 058.1.1992.116

Rosita Herawati,ST,MIT

NPP : 058.1.2004.263

Examiner

Hironimus Leong, S.Kom., M.Kom

NPP : 058.1.2007.273

Supervisor,

Dean of Faculty of Computer Science,

Robertus Setiawan Aji,ST,MCompIT

NPP :058.1.2004.264

Hironimus Leong, S.Kom., M.Kom

NPP : 058.1.2007.273

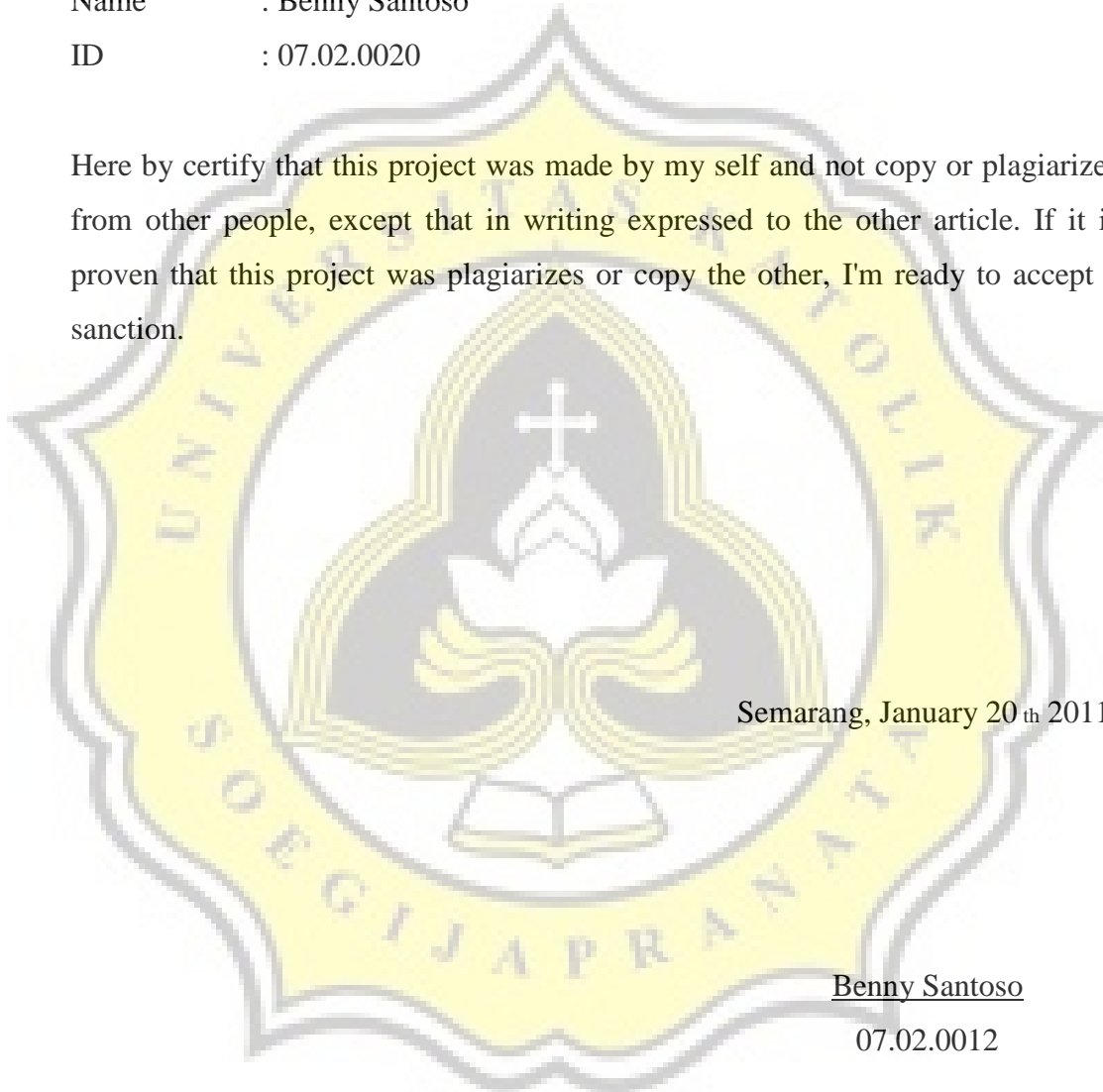
# STATEMENT OF ORIGINALITY

Hereby signed :

Name : Benny Santoso

ID : 07.02.0020

Here by certify that this project was made by my self and not copy or plagiarizes from other people, except that in writing expressed to the other article. If it is proven that this project was plagiarizes or copy the other, I'm ready to accept a sanction.



Semarang, January 20<sup>th</sup> 2011

Benny Santoso

07.02.0012

# FOREWORD

Thanks to God: Father, Son, and Holy Ghost for the bless, I have been completed this project with title: Implementation Syntactic Analysis and Semantic Analysis in Building Own Interpreter.

In this opportunity, I would thank to :

1. My Lord and my saviour, Jesus Christ that give me faith and courage to finish this project.
2. My parents, Agustono and Lie Miauw Lian and my sister Sari Indriani for their support, love, and pray.
3. Robertus Seiawan Aji N,ST., McomIT as my supervisor, for his advice, and ideas that inspired me.
4. All lecturers in Faculty of Computer Science, Hironimus Leong, S.Kom., M.Kom , Suyanto EA., Ir, M.Sc, Rosita Herawati ST.,MIT., Gregorius H. A. K., S.Si., M. Cs for their help and advice in programming.
5. All of my partner in RectMedia Group, Christian Chandra and Yohanes Venario and also all of my friends in IKOM for help and support me to finish this project,

Finally, I apologize because this project is not perfect, there still some bugs but I hope this project can inspire other students to make better projects. I'll receive any questions and critics about this project in open arm.

Semarang, January 20<sup>th</sup>, 2011

Benny Santoso

## ABSTRACT

Interpreter is a program for translates some form of source code into a target representation that it can immediately execute and evaluate. Where in this interpreter can interpret some of source code, which this code is with author coding style but still not much different from parsing in other programming language (like Java or PHP or etc). This interpreter build from two steps of analysis, namely Syntactic Analysis and Semantic Analysis and one step to parse to produces an output, namely Data Parsing. This interpreter is also use tree data structure, namely General Tree to save every break words into tokens.

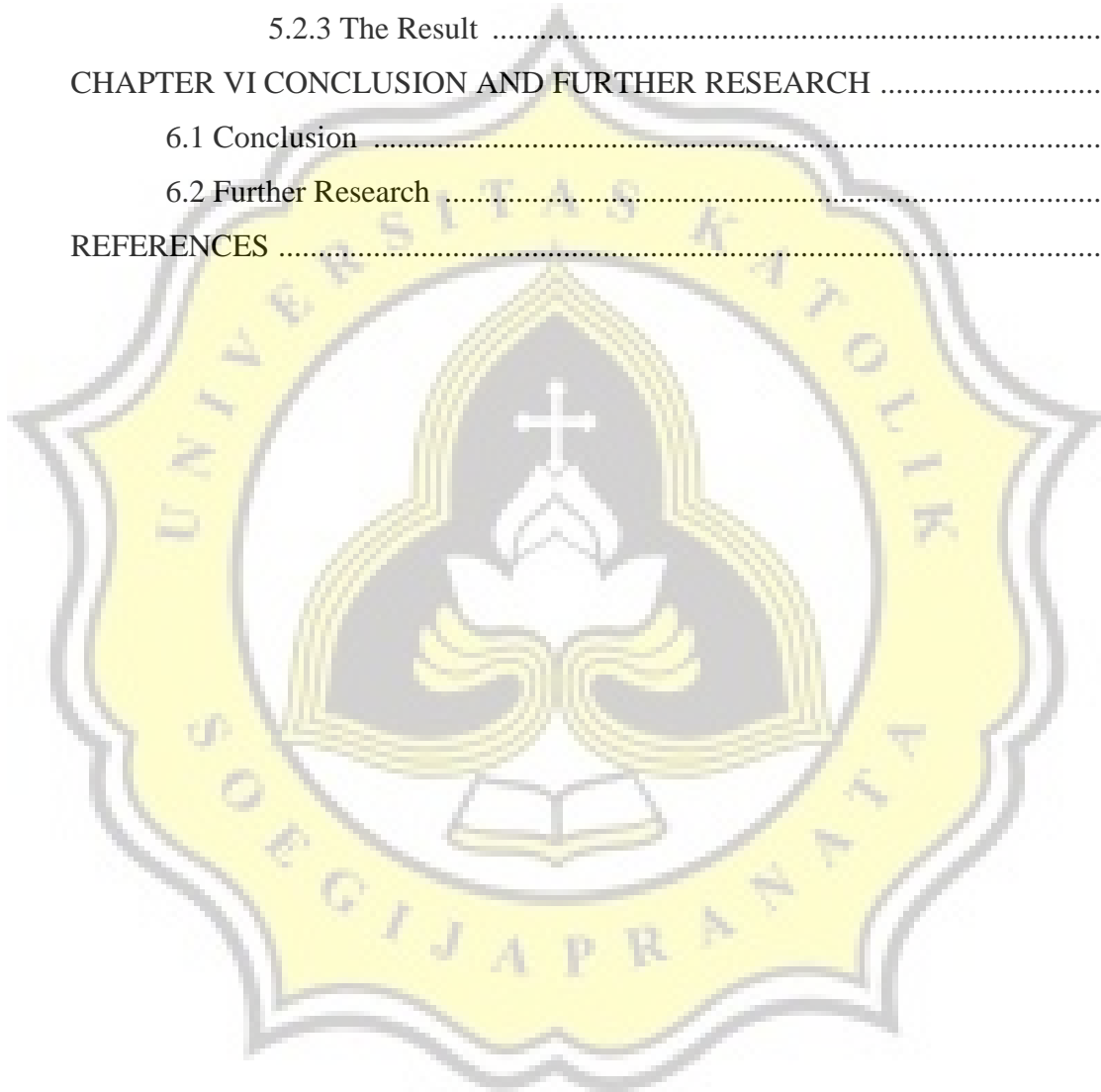
*Keyword: Syntactic Analysis, Semantic Analysis, and tokens*



## Table of Content

APPROVAL AND RATIFICATION PAGE .....	i
STATEMENT OF ORIGINALITY .....	ii
FOREWORD .....	iii
ABSTRACT .....	iv
Table of Content .....	v
Table of Figure .....	vii
Table of Table .....	viii
CHAPTER I INTRODUCTION .....	1
1.1 Background .....	1
1.2 Scope .....	1
1.3 Objective .....	2
CHAPTER II LITERATURE STUDY .....	3
2.1 Data Structure .....	3
2.1.1 Interpreter .....	3
2.2 Algorithm .....	3
2.2.1 Syntactic Analysis .....	3
2.2.2 Semantic Analysis .....	5
2.2.3 Parsing Data .....	6
CHAPTER III PLANNING .....	8
3.1 Research Methodologies .....	8
3.2 Project Management .....	8
CHAPTER IV ANALYSIS AND DESIGN .....	9
4.1 Analysis .....	9
4.1.1 Use Case Diagram .....	9
4.2 Class Diagram .....	7
4.2.1 Communication Class Diagram .....	10
4.2.2 Class Diagram Details .....	12

CHAPTER V IMPLEMENTATION AND TESTING .....	21
5.1 Implementation .....	21
5.2 Testing .....	30
5.2.1 Write In source ben .....	30
5.2.2 Running Interpreter Program .....	31
5.2.3 The Result .....	31
CHAPTER VI CONCLUSION AND FURTHER RESEARCH .....	34
6.1 Conclusion .....	34
6.2 Further Research .....	34
REFERENCES .....	35



## Table of Figure

Figure 2.1 : Illustrates Algorithm in Interpreter .....	7
Figure 4.1 : Use Case Diagram .....	9
Figure 4.2 : Communication Class Diagram.....	10
Figure 4.2.1 : Class Interpreter.....	12
Figure 4.2.2 : Class ReadFile .....	12
Figure 4.2.3 : Class Tokenizer .....	13
Figure 4.2.4 : Class SyntaxValidator.....	14
Figure 4.2.5 : Class SemanticValidator.....	17
Figure 4.2.6 : Class TreeNode.....	18
Figure 4.2.7 : Class Node.....	18
Figure 4.2.8 : Class Vector .....	19
Figure 4.2.9 : Class Mapping .....	20
Figure 5.2.1 : Sample Source Code.....	30
Figure 5.2.2 : Running Program .....	31
Figure 5.2.3 : The Result 01 .....	31
Figure 5.2.4 : The Result 02 .....	32
Figure 5.2.5 : The Result 03 .....	33



## Table of Table

Table 2.1 : Tabel Tokenizer .....	5
Table 3.1 : Project Management .....	8

