IMPLEMENTATION OF REAL-TIME SCANNER JAVA LANGUAGE TEXT WITH MOBILE VISION ANDROID BASED

BACHELOR THESIS

arrangedby

Fariz Dzulfiqar Nurzam

14.61.0020

UNDERGRADUATE PROGRAM
BACHELOR OF INFORMATICS
FACULTY OF COMPUTER SCIENCE
UNIVERSITY OF AMIKOM YOGYAKARTA
YOGYAKARTA

2017

IMPLEMENTATION OF REAL-TIME SCANNER JAVA LANGUAGE TEXT WITH MOBILE VISION ANDROID BASED

BACHELOR THESIS

to meet most of requirements of achieving a Bachelor's degree
Study Program Informatics

arranged by

Fariz Dzulfiqar Nurzam

14.61.0020

UNDERGRADUATE PROGRAM
BACHELOR OF INFORMATICS
FACULTY OF COMPUTER SCIENCE
UNIVERSITY OF AMIKOM YOGYAKARTA
YOGYAKARTA
2017

APPROVAL

BACHELOR THESIS

IMPLEMENTATION OF REAL-TIME SCANNER JAVA LANGUAGE TEXT WITH MOBILE VISION ANDROID BASED

prepared and arranged by

Fariz Dzulfiqar Nurzam

14.61.0020

was approved by the Bachelor Thesis Supervisor

at the date of 29 August 2017

Supervisor,

Emha Taufiq Luthfi, S.T., M.Kom

NIK. 190302125

LEGALIZATION

BACHELOR THESIS

IMPLEMENTATION OF REAL-TIME SCANNER JAVA LANGUAGE TEXT WITH MOBILE VISION ANDROID BASED

prepared and arranged by

Fariz Dzulfiqar Nurzam

14.62.0020

has been maintained in front of the Board of Examiners
on 15 August 2017

Composition of the Board of Examiners

Examiners	Signature
Sudarmawan, S.T., M.T **NIK. 190302035**	
Barka Satya, M.Kom **NIK. 190302126**	
Emha Taufiq Luthfi, S.T., M.Kom **NIK. 190302125**	

This bachelor thesis has been accepted as one of the requirements
to achieve a Bachelor degree in Computer
Agust 29, 2017

DEAN FACULTY OF COMPUTER SCIENCE

Krisnawati, S.Si, M.T.

NIK. 190302038

STATEMENT

I the undersigned declare that this thesis is my own work (ORIGINAL), and the contents in this paper there are works that have been asked by others to obtain an academic degree in a higher education institution anywhere, and my knowledge is not there works or opinions ever written and / or published by others, except that in writing referred to in this text and mentioned in the bibliography.

Everything associated with the script and the work that has been made is the responsibility of my own.

Yogyakarta, 29 August 2017

Fariz Dzulfiqar Nurzam

NIM. 14.61.0020

MOTTO

“Postera Crescam Laude”

“We shall grow in the esteem of future generations”

DEDICATION

Bismillah hirrahman nirrahim, Alhamdulillahirrabil'alamin, thanks to Allah SWT who has given me grace, guidance, health, and ability, so I can finish this bachelor thesis. On this presentation page, I would like to say many thanks for:

1. My parent, Mr. Wahyu Nurzam, S.E and Mrs. Reni Pujiastuti, who raise me with love, happiness, knowledge, and always do the best for me. My dear parent,you always own my dedication, thank you everything you've done.
2. My sisters, Aghnia Ilmi Sadida Nurzam, S.E and Hanin Hanifah Nurzam, My childhood has been memorable, all because of a sister who is so lovable.
3. Much obliged for Mr. Asro Nasiri, Drs, M.Kom., Mrs. Eny Nurnilawati, S.E, M.M., and Mrs. Suyatmi, S.E, M.M. Thank you for your help and support during the recent unstable condition.
4. Mr. Emha Taufiq Luthfi, S.T.,M.Kom as my supervisor who has guided me all this time and help my bachelor thesis completed.
5. The one who stole my attention, Alfiannisa' Nurkholisotin Ni'mah, may Allah always on your side my lucky charm.
6. All the staff of Innovation Center's director, staff, and student staff,thanks for the time, work experience, motivation, support, and happiness. Thank you for the opportunity that is always given to me so I can continue to study especially in the field of programming.

7. All of Lecturers who have taught me at Amikom University from 2014-2017. Thank you for the knowledge that has been given to me. Hopefully you are always given health and protection by Allah SWT.
8. All my friends from International Class, Amikom University, and Startup Founders Community, who give their support for me.
9. All my software developers greatest team. Thank you for all of your dedication and hard work. May Allah always bless us.
10. All parties who have been involved either directly or indirectly in the process of completing this thesis which can not be mentioned one by one.

PREFACE

Praise and gratitude to Allah SWT who has given mercy, guidance and strength so that author can finish this undergraduate thesis in accordance with the desired time. Do not forget sholawat and greetings are extended to the esteemed Prophet Muhammad SAW, which has been spreading Islam so that all Muslims can feel the beauty of Islam.

This undergraduate thesis is structured as a graduation requirement for all students at the University of Amikom Yogyakarta. It is also a proof that the student has finished undergraduate level and to obtain a Bachelor's degree in Computer.

With the completion of this undergraduate thesis, the author wants to thank to:

1. Prof. Dr. M. Suyanto, MM. as The Rector of University of Amikom Yogyakarta.
2. Mr. Emha Taufiq Luthfi, S.T., M.Kom as the supervisor who has provided guidance and assistance to author in the working of undergraduate thesis.
3. Author’s parents who always provided support and prayers.
4. Mr. and Mrs. Lecturers at University of Amikom Yogyakarta, which hasprovided knowledge for author lectures.

Yogyakarta, August 29, 2017

Author

TABLE OF CONTENTS

TITLE ii
APPROVAL iii
LEGALIZATION iv
STATEMENT v
MOTTO vi
DEDICATION vii
PREFACE ix
TABLE OF CONTENTS x
LIST OF TABLES xv
LIST OF FIGURES xvi
LIST OF APPENDICES xviii
INTISARI xix
ABSTRACT xx
CHAPTER I INTRODUCTION 1
1.1. BACKGROUND 1
1.2. PROBLEM STATEMENT 3
1.3. SCOPE OF PROBLEM 3
1.4. PURPOSE AND OBJECTIVE 4
1.5. RESEARCH BENEFIT 5
1.6. METHOD OF RESEARCH 6
1.6.1. Study Review 6
1.6.2. Analysis 6
1.6.3. Design 7
1.6.4. Implementation 7
1.6.5. Testing 7
1.7. SYSTEMATICS OF WRITING 7
1.7.1. Chapter I :Introduction 8
1.7.2. Chapter II :Theoretical Basis 8
1.7.3. Chapter III :Analysis and Design 8

1.7.4. Chapter IV :Implementation and Testing 8
1.7.5. Chapter V :Discussion 8
CHAPTER II THEORETICAL BASIS 9
2.1. LITERATURE REVIEW 9
2.2. BASIC THEORY 10
2.2.1. Android 10
2.2.2. Android Architecture 11
2.2.2.1. Linux Kernel 11
2.2.2.2. Libraries 11
2.2.2.3. Android Runtime 12
2.2.2.4. Applications Framework 13
2.2.2.5. Applications 13
2.2.3. Android Application Life Cycle 13
2.2.3.1. onCreate() Method 14
2.2.3.2. onStart() Method 14
2.2.3.3. onRestart() Method 14
2.2.3.4. onResume() Method 15
2.2.3.5. onPause() Method 16
2.2.3.6. onStop() Method 16
2.2.3.7. onDestroy() Method 16
2.2.4. Google Mobile Vision 16
2.2.5. Optical Character Recognition 17
2.2.5.1. Optical Scanning 18
2.2.5.2. Location Segmentation 18
2.2.5.3. Preprocessing 18
2.2.5.4. Future Extraction 18
2.2.5.5. Post-Processing 19
2.2.6. Restful Web Services 19
2.2.7. JSON 20
2.2.8. Codeigniter as builder of RESTweb service 21

CHAPTER III ANALYSIS AND DESIGN 22
3.1. ARCHITECTURE OF MOBILE VISION TEXT PROCESSING APPLICATION CONCEPT 22
3.2. NEEDS ANALYSIS 23
3.2.1. Functional Needs Analysis 23
3.2.2. Non-functional Needs Analysis 24
3.2.2.1. Operational 24
3.2.2.2. Security 25
3.2.2.3. Information 25
3.2.2.4. Performance 25
3.3. ARCHITECTURE DESIGN OF MOBILE VISION TEXT PROCESSING APP 25
3.4. USER INTERFACE DESIGN 26
3.5. UNIFIED MODELING LANGUAGE DESIGN 27
3.5.1. Use Case Diagram Design 27
3.5.2. Activity Diagram Design 31
3.5.4. Class Diagram Design 36
3.5.5. Sequence Diagram Design 38
CHAPTER IV IMPLEMENTATION AND TESTING 44
4.1. IMPLEMENTATION 44
4.1.1. Web Service Database Implementation 44
4.1.1.1. Create Web Service Database 44
4.1.1.2. Create Web Service Database Table 46
4.1.1.3. Create Web Service Database Dumping Data 47
4.1.2. Implementation of Web Service Application Architecture 47
4.1.2.1. Upload of Web Service Application 47
4.1.2.2. Basic Configuration of Web Service Appication 49
4.1.2.3. Configuration of Web Service Database 50
4.1.2.4. Implementation of Library REST Controller 50
4.1.2.5. Create Script Model of Web Service Application 51

4.1.2.6. Create Script Controller of Web Service Application 51
4.1.2.7. Create Documentation of Web Service Application 52
4.1.3. Implementation of Android App Architecture 55
4.1.3.1. Create Basic Android Application 55
4.1.3.2. Implementation of Mobile Vision Library..... 58
4.1.3.3. Implementation of Volley Library 58
4.1.3.4. Create Script of Android Layout 59
4.1.3.5. Create Script of Android Camera..... 60
4.1.3.6. Create Script of Check Internet Connection 61
4.1.3.7. Create Script of Android API Request 62
4.1.3.8. Create Script of Android Text Processing 65
4.2. TESTING..... 67
4.2.1. Test Database User with Login 67
4.2.2. Test Record Data with Query Select..... 68
4.2.3. Web Service Test with Postman..... 69
4.2.4. Camera Test in Android Text Processing Application 71
4.2.5. Internet Connection Test in Android Text Processing Application..... 72
4.2.6. Text Processing Test in Android Text Processing Application 73
4.2.7. Translate Two Language Test in Android Text Processing Application..... 74
4.2.8. Test of Generate Documentation 75
4.2.9. Test of Build and Generate Signed APK 77
4.3. DISCUSSION 80
4.3.1. Text Processing Flow..... 80
4.3.1.1. Scanning Text in Frame Process 80
4.3.1.2. Process of HTTP Request with Volley..... 82

4.3.1.3. Process of Query and Handling in Web Service App 83
4.3.1.4. Volley Callback Process 85
4.3.1.5. Process of Display the Result Text Processing 86
4.3.2. Installation and Test Result 87
4.3.3. Advantages and Disadvantages 93
CHAPTER V DISCUSSION 94
5.1. CONCLUSION 94
5.2. SUGGESTION 95
BIBLIOGRAPHY 96
APPENDIX 1

LIST OF TABLES

Table 3.1 Design details of use case diagram 29
Table 3.2 Activity diagram details of use case scan text in camera frame 31
Table 3.3 Activity diagram details of use case get text data 32
Table 3.4 Activity diagram details of use case process input data 33
Table 3.5 Activity diagram details of use case diagram response output in JSON 34
Table 3.6 Activity diagram details of use case diagram response show translated text 35
Table 4.1 Testing devices comparation 90
Table 4.2 Test result of testing devices..... 90

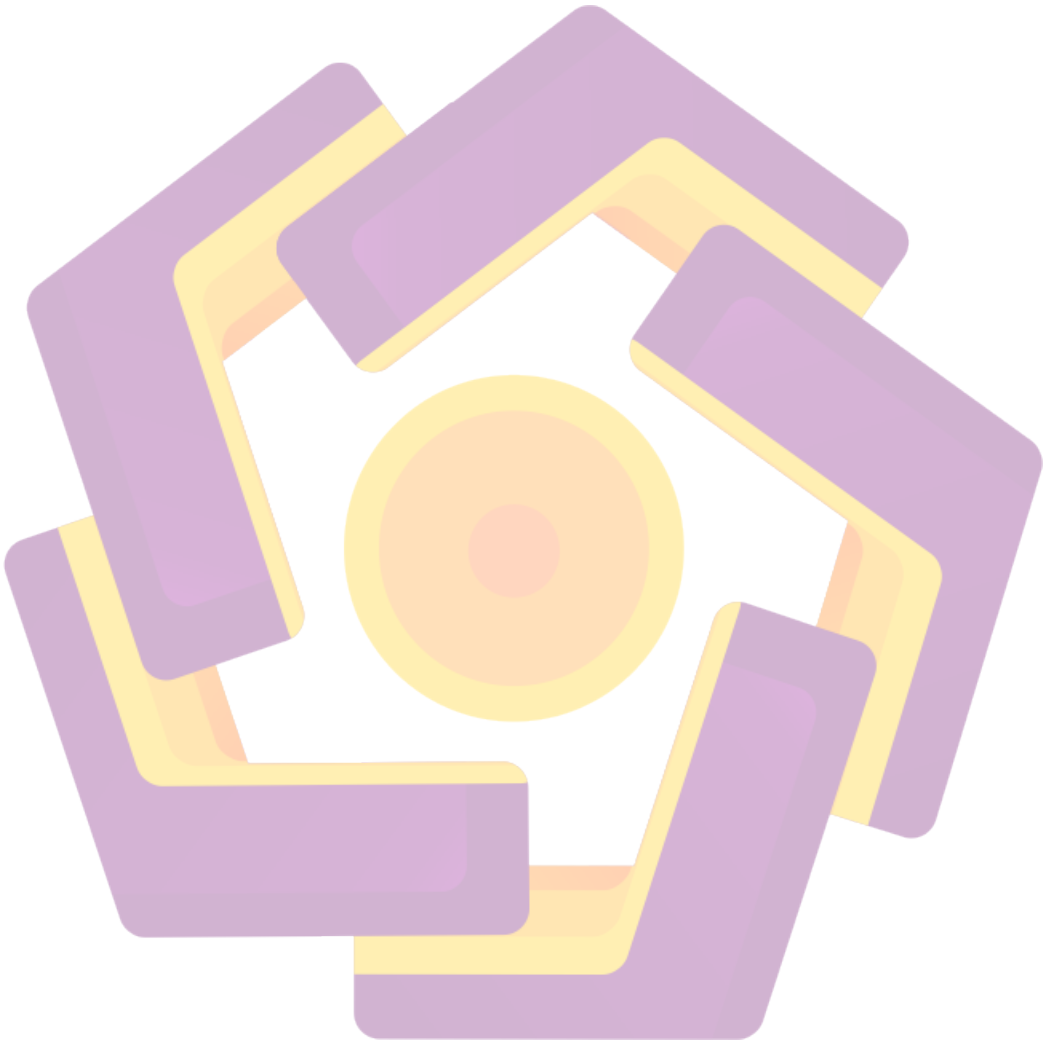
LIST OF FIGURES

Figure 2.1 Android architecture stack 12
Figure 2.2 Android application life cycle 15
Figure 2.3 Optical character recognition components 17
Figure 2.4 RESTful web service work flow 20
Figure 2.5 JSON sctructure 21
Figure 3.1 Architecture design of text processing app 26
Figure 3.2 Application wireframe design 27
Figure 3.3 Use case diagram design 28
Figure 3.4 Activity diagram design of use case scan text in camera frame 31
Figure 3.5 Activity diagram design of use case get text data 32
Figure 3.6 Activity diagram design of use case process input data 33
Figure 3.7 Activity diagram design of use case diagram response output in JSON 34
Figure 3.8 Activity diagram design of use case diagram response show translated text 35
Figure 3.9 Class diagram design of Android app 36
Figure 3.10 Class diagram design of web service app 37
Figure 3.11 Sequence diagram design of use case scan text in camera frame 39
Figure 3.12 Sequence diagram design of use case get text data 40
Figure 3.13 Sequence diagram design of use case process input data 41
Figure 3.14 Sequence diagram design of use case diagram response output in JSON 42
Figure 3.15 Sequence diagram design of use case diagram response show translated text 43
Figure 4.1 Create new database 44
Figure 4.2 Create new database user 45
Figure 4.3 Add new database user 46
Figure 4.4 File Manager 48
Figure 4.5 Success upload file 49

Figure 4.6 Success extraction of Codeigniter application 49
Figure 4.7 Android Studio – New project 56
Figure 4.8 Android Studio – Target Android devices 56
Figure 4.9 Android Studio – Add an activity to mobile 57
Figure 4.10 Android Studio – Customize the activity 57
Figure 4.11 PhpMyAdmin login page 67
Figure 4.12 Result of user test in database 68
Figure 4.13 Result of first query test 69
Figure 4.14 Result of second query test 69
Figure 4.15 Result of Postman with Indonesian word 70
Figure 4.16 Result of Postman with Javanese word 71
Figure 4.17 Result of Postman with random word 71
Figure 4.18 Result of camera test in Android app 72
Figure 4.19 Result of internet connection test in Android app 73
Figure 4.20 Result of text processing test in Android app 74
Figure 4.21 Result of two language translate in Android app 75
Figure 4.22 Execution result of phpDoc command 76
Figure 4.23 Generate result of phpDoc documentation 77
Figure 4.24 Result of build APK test 78
Figure 4.25 Dialog window of New Key Store 79
Figure 4.26 Dialog window of Generate Signed APK 79
Figure 4.27 Result of Geneate Signed APK 80
Figure 4.28 Android app installation 88
Figure 4.29 Success message of Android app installation 89
Figure 4.30 Test result with testing device A 91
Figure 4.31 Test result of testing device B 92

LIST OF APPENDICES

Appendix A Repository Git Project.....1



INTISARI

Saat ini penelitian teknologi di bidang alih bahasa telah merambah hingga pengolahan citra.Salah satu peran pengolahan citra dalam bidang alih bahasa adalah teknologi pengenalan karakter optik pada berbagai perangkat, salah satunya berbasis mobile Android. *Namun, saat ini penelitian pengolahan citra di bidang alih bahasa dengan bahasa daerah asli Indonesia sangat sedikit*.

Dalam penelitian ini, penulis melakukan implementasi pemindai teks latin dengan Bahasa Indonesia menjadi teks Bahasa Jawa dengan Google Mobile Vision *secara* real-time *dan sebaliknya dengan aplikasi berbasis* mobile Android. *Dimana* Google Mobile Vision *merupakan salah satu pustaka* optical character recognition *yang bersifat* open-source *dengan metode* pattern recognition. *Dengan rancangan arsitektur* client server *maka dikembangkan aplikasi Android sebagai* client *dan aplikasi* Web Service *yang dapat mengolah* REST API *dan berinteraksi dengan* database.

Luaran penelitian ini berupa aplikasi alih bahasa yang dapat memindai kata dalam Bahasa Jawa menjadi Bahasa Indonesia, maupun sebaliknya dalam suatu frame *aplikasi* Android. *Dengan penelitian ini diharapkan dapat menjadi* benchmark *untuk penelitian pengolahan citra untuk bahasa daerah asli Indonesia*.

Kata Kunci : *Pengenalan Karakter Optik, Mobile Vision, Android, Arsitektur Client Server, REST API*

ABSTRACT

Currently technological research in the translating has penetrated to image processing. One of the role of image processing in the translating is optical character recognition technology on various platform, an example on Android mobile. However, the current research on image processing of translating with the native regional language of Indonesia still rarely.

In this research, the writer performs the implementation of Latin text translate from Bahasa Indonesia into Javanese text with Google Mobile Vision in real-time, also vice versa with Android mobile based application. Which is Google Mobile Vision is one of the open-source optical character recognition library with pattern recognition method. With the design of client server architecture, the writer develop Android application as client and Web Service application that can process REST API and interact with the database.

The output of this research is an application for translate that can scan words in Javanese into Bahasa Indonesia, or vice versa in frame of an Android application. With this research, it's expected to be the benchmark for image processing research for native regional language of Indonesia.

Keywords :Optical Character Recognition, Mobile Vision, Android, Client Server Architecture, REST API