LIBRARY TESSERACT UTILIZATION ON APPKUNTAN APPLICATION ANDROID BASED

Lugy Septian Catur Firmansyah

Universitas Komputer Indonesia
Jalan Dipatiukur No. 112-116 Kota Bandung, Jawa Barat 40132
E-mail : lugyscf@gmail.com

ABSTRACT

The technology is rapidly growing at this time, many people need technology that can meet the needs of the right. From the previous technology originated from the PC or computer to the smartphone is rapidly growing at this time. As a personal financial manager with use their Tesseract library contained on the android Smartphone.

The library is a Tesseract machine optical character-recognition on android. That lets the user can make financial expenditures in regulating private spending, and can also help to provide a reminder to the problem of financial expenditure of personal shopping by taking a picture of a stroke using a camera shopping android smartphone which is then scanned with shopping data using tesseract in android smartphone, so data shopping which has been scanned will be processed by tesseract then the image data that has been processed will be displayed to the screen as the data output of text text data, which then may be stored in the database and will provide notification when financial expenses exceeding the maximum limit which has been specified earlier.

Keywords : Tesseract Library, Manage Finance, Android.

1. INTRODUCTION

1.1 Background

Financial management in everyday life is one of the main keys to undergo any necessities of life, starting with school children, students, workers, let alone to everyone who is already married, everything required to manage their finances properly. When uncontrolled financial usage then it would be a futile waste, which must be very self-defeating.

The reality in terms of finance, there are still many of us who could not control it, because most people unknowingly will buy something not too required only because just "wants", even though the goods bought it was pretty expensive price. So we certainly never thought "where is our money gone?" that's what will cause people to forget against his needs, and in the end we will regret it once sober.

Based on data obtained from a questionnaire that has been disseminated via social media, apparently from the 86 responses, which have the android smartphone is 89.5% or 9 people who don't have the android smartphone. And proven out of 77 responses from the question "have you ever been confusion where your money is gone?" 59.7% answer sometimes, 31.2% answered frequently, and 9.1% answered never. Which means most people who fill out questionnaires cannot control its financial reserves. It can be proven from the question "what is your financial expenses often uncontrolled?" and from the 77 responses 55.8% answer sometimes, 41.6% answered frequently, and 2.6% answered never.

Why everyone does not record his spending so that it can help to manage his personal finances? It is because the financial outlay in noting a bit difficult and also most people are lazy to take down its expenditure. In addition people's difficulty in entering data expenditures the Financials, so if in entering.

The data is taken through the image then can be facilitated by using library tesseract. How do I place the financial expenditure data are taken through the image? That is to take a picture of a stroke using a camera shopping android smartphone which is then scanned with shopping data using tesseract in android smartphone.

So the author gives suggestion for making the application helper that can alert the user of the financial expenditure. This application can take text from an image as an image stroke shopping using library tesseract machine i.e. optical character-recognition on android. If text data already obtained data may be processed, as it may give notice of personal financial expenditure. The application will read a character before a stroke in shopping photos and scanned, and then transformed into text data so the application will process data to provide notification of our shopping expenditure on every week or every month depending on its settings. How to use this application, we photographed the stroke enough shopping to get a reminder of the financial expenses at a later time, or if we are shopping in a place that doesn't give stroke we can enter data manually purchased items and enter the total expenditures incurred are manually as well.

And the making of the application based on the 77 responses from questionnaires which researchers ask: "do you agree with the rendered application android smartphone that can help remind your financial problems?" 42.9% replied strongly agree, 41.6% answered agree, 11.7% responded neutral, and 3.9% replied strongly disagree.

With this author took the title of the thesis "Library Tesseract Utilization On Appkuntan Application Android Based".

1.2 Problem Identification

Based on the background that has been expressed above, there is some problem, namely :

a. How do I apply for a Appkuntan who can provide a reminder to arrange finance users and also provide an ease in noting the financial expenditure.
b. how to utilize Library Tesseract on application Appkuntan so that it can facilitate each user process data within a stroke of shopping.

1.3 Goal and purpose

Based on the problems examined then the intent of the research is to produce an application appkuntan that help finance yourself android based.

While the goal will be achieved in this research is the application of appkuntan can make it easier for users to record the financial outlay by way of scanning stroke shopping using the tesseract, thus text data obtained by application of appkuntan can be in sports to remind the financial expenditure expenditure users.

1.4 Theory Broke Character Image Using Tesseract

The theory solves the character image using tesseract on android smartphones. In solving image character is making a good algorithm so that android smartphones can read or recognize any characters that exist in the image such as the man who did it.

The characters of the image need to be selected in advance to see if kertasnya is in good shape, color kertasnya color white with black ink and clear, there are no letters that mutually overwrite, paper is not terlipat-lipat or damaged. Thus, the tesseract can recognize text on the picture so well that got better results anyway.

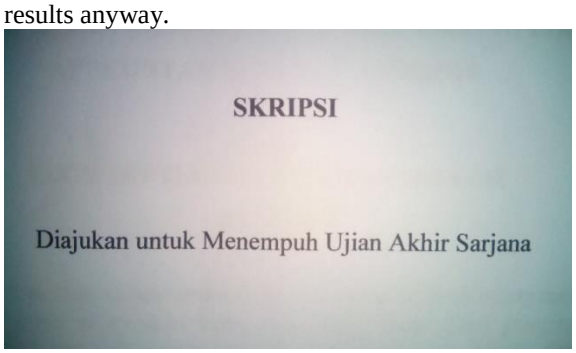


Figure 1 Examples Of Character Images

Here are the steps to troubleshoot data stroke using tesseract on android smartphone :

1. Take A Picture

Take pictures using the camera android smartphone or by taking a picture from the gallery.

2. Cutting Pictures (Cropping)

After the picture is taken or chosen then goes into the process of cutting the images. Cutting can be done by way of menggeserkan any point cut, so that the obtained data that has been desired. See sample picture below :

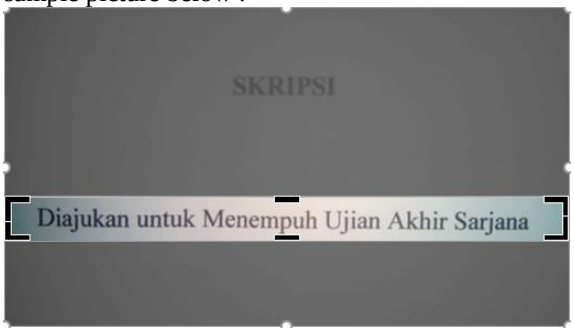


Figure 2 Examples Of Character Images Cutting

3. The Process Of Character Recognition

After the cutting results are stored then android smartphone system will do the process of character recognition using tesseract. Where once the character images successfully read or recognized then the system will display the results of the character data of the image. Coding process for tesseract character recognition on images to display the results of the data that has been read or recognizable, namely :

```
private void prepareTesseract() {
      try {
         prepareDirectory(DATA_PATH + 
TESSDATA);
      } catch (Exception e) {
         e.printStackTrace();
      }
      copyTessDataFiles(TESSDATA);
   }

private void copyTessDataFiles(String path) {
      try {
         String fileList[] = 
getApplicationContext().getAssets().list(path);

         for (String fileName : fileList) {

            // open file within the assets folder
            // if it is not already there copy it to 
the sdcard
            String pathToDataFile = 
DATA_PATH + path + "/" + fileName;
            if (!(new 
File(pathToDataFile)).exists()) {

               InputStream in = 
                  getApplicationContext().getAsset
                  s().open
```

```
                (path + "/" + fileName);

            OutputStream out = new
FileOutputStream(pathToDataFile);

            // Transfer bytes from in to out
            byte[] buf = new byte[1024];
            int len;

            while ((len = in.read(buf)) > 0) {
               out.write(buf, 0, len);
            }
            in.close();
            out.close();

            Log.d(TAG, "Copied " + fileName
+ "to tessdata");
          }
        }
      } catch (IOException e) {
        Log.e(TAG, "Unable to copy files to
tessdata " + e.toString());
      }
   }

private void startOCR() {
      try {
        BitmapFactory.Options options = new
BitmapFactory.Options();
        options.inSampleSize = 4; // 1 - means
          max size. 4 - means maxsize/4
          size. Don't use value <4, because you
          need more memory in the heap
          to store your data.
        //Bitmap bitmap =
          BitmapFactory.decodeFile(imgUri.get
          Path(), options);

        result = extractText(thePic);
        textView.setText(result);
        parsingText();

      } catch (Exception e) {
        Log.e(TAG, e.getMessage());
      }
   }

private String extractText(Bitmap bitmap) {
      try {
        tessBaseApi = new TessBaseAPI();
      } catch (Exception e) {
        Log.e(TAG, e.getMessage());
        if (tessBaseApi == null) {
          Log.e(TAG, "TessBaseAPI is null.
            TessFactory not returning tess
            object.");
        }
      }

      tessBaseApi.init(DATA_PATH, lang);
```

```
//      //EXTRA SETTINGS
//        //For example if we only want to detect
numbers
//
          tessBaseApi.setVariable(TessBaseAPI.V
          AR_CHAR_WHITELIST,
          "1234567890");
//
//       //blackList Example

        //tessBaseApi.setVariable(TessBaseAPI.V
        AR_CHAR_BLACKLIST,
        "!@#$%^&*()_       +=-,./<>?");

      Log.d(TAG, "Training file loaded");
      tessBaseApi.setImage(bitmap);
      String extractedText = "empty result";
      try {
        extractedText                          =
tessBaseApi.getUTF8Text();
      } catch (Exception e) {
        Log.e(TAG, "Error in recognizing
text.");
      }
      tessBaseApi.end();
      return extractedText;
   }
```

1.5 Analysis Method

Here are the stages with the method used in the application to be built, from input to output.

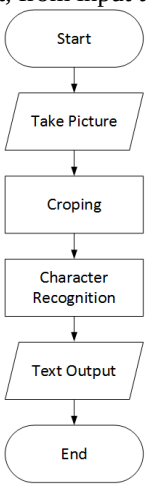


Figure 3 Stages On The Application

1. *Take Picture*

This stage is the early stages are used to take pictures through the camera that is mounted on an android device, as the insert will be processed by the next stage.

2. *Cropping*

This stage serves to cut out the pictures that have been taken by the user. Stage of the cropping necessary to focus the image on the text area so that no object other than text to be recognized by the OCR. Users define your own text on the image area to be cut, so that the image size smaller so as to

accelerate the process of penengenalan characters. The process of cutting the images using the functions that are already on your android device.

3. *Character Recognition*

The process of character recognition using a service that already exists, namely the Tesseract.

4. *Text Output*

The results of the introduction of the character by OCR will be sent back to the application in the form of text files that will be stored into the database.

1.6 Analysis Of The Process Of Parsing The Data A Stroke

To be able to generate the appropriate data format is stroke shopping is required the process of parsing that is parsing every word or numbers on a stroke of shopping. Coding for decomposition of characters on a picture of a stroke to display the results of the data that has been read or recognizable, namely :

```
private void parsingText() {
      StringTokenizer tokens = new StringTokenizer (textView.getText().toString(), "\r\n");
      if (tokens.countTokens() == 1 ){
         String satu = tokens.nextToken();
         String harga1 = satu.substring(satu.length() - 5);
         String hargasat1 = satu.substring(satu.length() - 10);
         String hargasat1_1 = hargasat1.substring(0, 4);
         String qty = satu.substring(satu.length() - 12);
         String qty_1 = qty.substring(0, 1);

         satu = satu.replaceAll("[\\d.,]","");
         harga1 = harga1.replaceAll("[.,]","");

         aq.id(R.id.item1).text(satu);

aq.id(R.id.item2).visibility(View.GONE);

aq.id(R.id.item3).visibility(View.GONE);

aq.id(R.id.item4).visibility(View.GONE);

aq.id(R.id.item5).visibility(View.GONE);

aq.id(R.id.containerHasil).visibility(View.VISIBLE);

aq.id(R.id.textResult).visibility(View.GONE);

         aq.id(R.id.qty1).text(qty_1);
         aq.id(R.id.qty2).visibility(View.GONE);
         aq.id(R.id.qty3).visibility(View.GONE);
         aq.id(R.id.qty4).visibility(View.GONE);
         aq.id(R.id.qty5).visibility(View.GONE);

aq.id(R.id.containerQty).visibility(View.VISIBLE);

aq.id(R.id.textResultqty).visibility(View.GONE);

         aq.id(R.id.hrgitem1).text(harga1);

aq.id(R.id.hrgitem2).visibility(View.GONE);

aq.id(R.id.hrgitem3).visibility(View.GONE);

aq.id(R.id.hrgitem4).visibility(View.GONE);

aq.id(R.id.hrgitem5).visibility(View.GONE);

aq.id(R.id.containerHasilHrg).visibility(View.VISIBLE);

aq.id(R.id.textResultHrg).visibility(View.GONE);

         aq.id(R.id.hrg2item1).text(hargasat1_1);

aq.id(R.id.hrg2item2).visibility(View.GONE);

aq.id(R.id.hrg2item3).visibility(View.GONE);

aq.id(R.id.hrg2item4).visibility(View.GONE);

aq.id(R.id.hrg2item5).visibility(View.GONE);

aq.id(R.id.containerHasilHrg2).visibility(View.VISIBLE);

aq.id(R.id.textResultHrg2).visibility(View.GONE);
```

1.7 A Functional Needs Analysis

Table 1 Software Functional specifications on Mobile platforms

Functional Software Requirements Specification on the Mobile Platform	
Name	**Description**
SKPL-F-001	The system provides on-site login
SKPL-F-002	The system provides facilities take with the camera
SKPL-F-003	The system provides facilities take from galleries
SKPL-F-004	The system provides facilities settings
SKPL-F-005	The system provides facilities enter the manual
SKPL-F-006	The system provides data on shopping
SKPL-F-007	The system provides facilities to send data via web services
SKPL-F-008	The system provides facilities to receive data via web services

SKPL-F-009	The system provides facilities create an account
SKPL-F-010	The system provides facilities logout

Table 2 Specification of Non Functional Software in Mobile platforms

Software Requirements Specification is a Non Functional on Mobile platforms	
Name	**Description**
SKPL-NF-001	The system can be accessed anytime anywhere
SKPL-NF-002	The android-based systems
SKPL-NF-003	The system can run with a minimum android version 4.4.2 (Kit Kat)
SKPL-NF-004	The system can run optimally on Smartphones that have the above camera 8 MP

1.8 Use Case Diagram

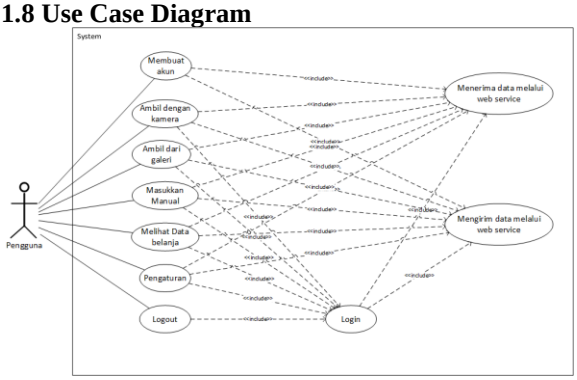


Figure 4 Use Case Diagram

1.9 Diagram Of Relation

Diagram of relation is the set of relationships between two or more tables in a database system. A picture Diagram of the Relationship is the image of a series of databases on development Appkuntan.

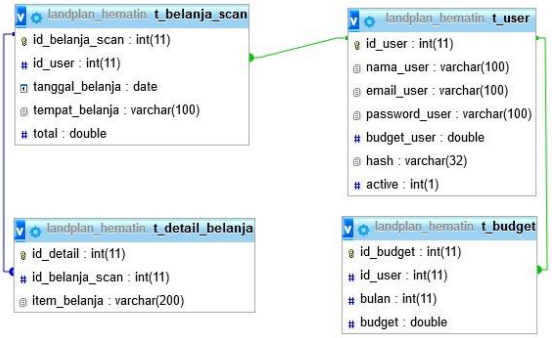


Figure 5 Diagram Of Relation

1.10 Table Structure

Describing table structure that contains detail fields, data types, the length of the data, and other information. The tables contained in the data base, namely t_user, t_budget, t_belanja_scan, t_detail_belanja.

Table 3 Table Structure t_user

Field Name	Data Type	Length	Null	Extra	Key
id_user	Int	11	No	Auto increment	Primary key
nama_user	Varchar	100	No		
email_user	Varchar	100	No		
password	Varchar	100	No		

Table 4 Table Structure t_budget

Field Name	Data Type	Length	Null	Extra	Key
id_budget	Int	11	No	Auto increment	Primary key
id_user	Int	11	No		Foreign key references t_user (id_user)
bulan	Int	11	No		
budget	double		No		

Table 5 Table Structure t_belanja_scan

Field Name	Data Type	Length	Null	Extra	Key
id_belanja_scan	Int	11	No	Auto increment	Primary key
id_user	Int	11	No		Foreign key references t_user (id_user)
tanggal_belanja	date		No		
tempat_belanja	varchar	100	No		

Table 6 Table Structure t_detail_belanja

Field Name	Data Type	Length	Null	Extra	Key
id_detail	Int	11	No	Auto increment	Primary key
id_belanja_scan	Int	11	No		Foreign key references t_belanja_scan (id_belanja_scan)
item_belanja	varchar	200	No		

Field Name	Data Type	Length	Null	Extra	Key
total_harga	Int	11	No		
jumlah	Int	11	No		
harga	int	11	No		

1.11 Menu Structure The Android Platform

The design of the structure of the menu is the description of the line usage application so that applications that are easy to understand and easy to use.

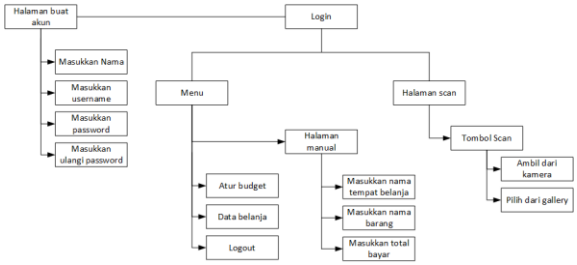


Figure 6 Menu Structure The Android Platform

1.12 The Hardware Implementation

The hardware that is used to build the system in a mobile platform for application Appkuntan as Assistant personal finance can be seen in the following table.

Figure 7 Minimum Hardware implementations in Mobile platforms

The Name Of The Hardware	**Specifications**
Proccessor	Quad-core 1.3 GHz Cortex-A7
Memory Internal	8 GB
Rear camera	8 MP
Screen	4.7 inches
The Operating System	Android OS, v4.2 (Jelly Bean)

1.13 Implementation Of Database

The creation of the database is done using the MySQL DBMS applications.

Table 8 Implementation Of Database

No	The Table Name	The SQL Command
1	t_belanja_scan	CREATE TABLE IF NOT EXISTS `t_belanja_scan` (`id_belanja_scan` int(11) NOT NULL, `id_user` int(11) NOT NULL, `tanggal_belanja` date NOT NULL, `tempat_belanja` varchar(100) NOT NULL, `total` double NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1; ALTER TABLE `t_belanja_scan` ADD PRIMARY KEY (`id_belanja_scan`), ADD KEY `t_belanja_scan_ibfk_1` (`id_user`); ALTER TABLE `t_belanja_scan` ADD CONSTRAINT `t_belanja_scan_ibfk_1` FOREIGN KEY (`id_user`) REFERENCES `t_user` (`id_user`) ON DELETE CASCADE ON UPDATE CASCADE;
2	t_budget	CREATE TABLE IF NOT EXISTS `t_budget` (`id_budget` int(11) NOT NULL, `id_user` int(11) NOT NULL, `bulan` int(11) NOT NULL, `budget` double NOT NULL) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=latin1; ALTER TABLE `t_budget` ADD PRIMARY KEY (`id_budget`), ADD KEY `t_budget_ibfk_1` (`id_user`); ALTER TABLE `t_budget` MODIFY `id_budget` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=14; ALTER TABLE `t_budget` ADD CONSTRAINT `t_budget_ibfk_1` FOREIGN KEY (`id_user`) REFERENCES `t_user` (`id_user`) ON DELETE CASCADE ON UPDATE CASCADE;
3	t_detail_belanja	CREATE TABLE IF NOT EXISTS `t_detail_belanja` (`id_detail` int(11) NOT NULL, `id_belanja_scan` int(11) NOT NULL, `item_belanja` varchar(200) NOT NULL) ENGINE=InnoDB AUTO_INCREMENT=118 DEFAULT CHARSET=latin1; ALTER TABLE `t_detail_belanja` ADD PRIMARY KEY (`id_detail`), ADD KEY `t_detail_belanja_ibfk_1` (`id_belanja_scan`); ALTER TABLE

		`t_detail_belanja` MODIFY `id_detail` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=118; ALTER TABLE `t_detail_belanja` ADD CONSTRAINT `t_detail_belanja_ibfk_1` FOREIGN KEY (`id_belanja_scan`) REFERENCES `t_belanja_scan` (`id_belanja_scan`) ON DELETE CASCADE ON UPDATE CASCADE;
4	t_user	CREATE TABLE IF NOT EXISTS `t_user` (`id_user` int(11) NOT NULL, `nama_user` varchar(100) NOT NULL, `email_user` varchar(100) NOT NULL, `password_user` varchar(100) NOT NULL, `budget_user` double NOT NULL, `hash` varchar(32) NOT NULL, `active` int(1) NOT NULL DEFAULT '0') ENGINE=InnoDB AUTO_INCREMENT=20 DEFAULT CHARSET=latin1; ALTER TABLE `t_user` ADD PRIMARY KEY (`id_user`); ALTER TABLE `t_user` MODIFY `id_user` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=20;

1.14 Implementation Of Interface

Implementation of the interface of each program and display pengkodeannya in the form of program files.

Table 9 Implementation of Interfaces in Mobile platforms

Menu	Description	File Name
Login	Is the page to Login	activity_login.xml
The main menu	Is the start page when a user first login	activity_main.xml
Indomaret Page	Is the page that shows the page scans for stroke	activity_indomaret.xml
	Indomaret	
Scan Struk	Is the page to scan for struk	activity_scan.xml
Spending Data	Is the page to see your shopping data that has been scanned.	activity_data_belanja_scan.xml
Setting The Budget	Is the page that displays the settings for the maximum financial expenditure limit per month.	activity_budget.xml
Create An Account	Is the page to create a new account.	activity_daftar.xml

1.15 Conclusion

This is a conclusion which can be drawn on the final results of this research.

1. Appkuntan applications on the android platform is quite able to assist the user in performing the financial record-keeping.
2. This appkuntan application and also quite can remind users in the amount of expenses paid for.

1.16 Advice

The appkuntan software has been built already can solve the problem at hand. As for the advice against this appkuntan software development is as follows :

1. The lack of constraints on the ability of the library tesseract recognize handwriting tend not to have a special pattern.
2. Difficulty recognizing a stroke writing letters or overwrite each other she said.
3. Difficulty recognizing writing a letter or a stroke he said was unclear.
4. Recognizing the difficulty of writing a stroke if a stroke is not a white paper.
5. Unable to read the entire stroke.

REFERENCE

[1] Smith. Ray, "An Overview of Tesseract Engine," 2013.

[2] B. R. Sofian, "Perbandingan Algoritma Template Matching Dan Feature Extraction Pada Optical Character Recognition," vol. I, Bandung : s.n., 2012.

[3] K. Hamilton and R. Miles, Learning UML 2.0, O'Reilly, April 2006, p. 286.
[4] A. Solichin, MySQL 5 Dari Pemula Hingga Mahir, Jakarta: Achmatim.net, 2010.