

## Rancang Bangun Aplikasi Penerjemah Bahasa Jepang – Indonesia Menggunakan OCR Berbasis Android

Aulia Rizqi\*<sup>1</sup>, Nur Kasan<sup>2</sup>, Widiyanto<sup>3</sup>

<sup>1,2,3</sup> Fakultas Teknik, Jurusan Teknik Elektro, Universitas Muhammadiyah Malang  
Auliarizqi1010@gmail.com<sup>1</sup>

### Abstract

*Japanese kanji in a book or picture is very difficult to read, especially for people who are just learning Japanese. For people who are just learning Japanese to find out how to read and the meaning of a kanji need 2 different types of dictionaries, namely dictionaries for reading and dictionary for their meanings. To make it easier to find out how to read kanji letters and their meanings, researchers created an android application that can detect Japanese language kanji using a smartphone camera and displays the meaning and how to read it on a smartphone screen. In making this android application using java and xml programming on android studio. Then on android studio add a new tesseract OCR engine library and yandex translate API. Tesseract OCR engine serves as a detection of starch using a smartphone camera and yandex translate API functions as a Japanese translator into Indonesian. To get a way to read Japanese kanji, the researcher created a new database using SQLite which is used to change from kanji to alphabet. The application testing uses the black box testing method and obtains the accuracy of tesseract OCR engines by 87.5%.*

**Keywords:** Tesseract OCR Engine, Japanese Language, and Java.

### Intisari

Huruf kanji Bahasa Jepang yang ada pada sebuah buku atau gambar sangat sulit dibaca, terutama bagi orang yang baru belajar bahasa Jepang. Bagi orang yang baru belajar bahasa Jepang untuk mengetahui cara membaca dan arti dari sebuah huruf kanji membutuhkan 2 jenis kamus yang berbeda, yaitu kamus untuk cara membaca dan kamus untuk artinya. Untuk mempermudah mengetahui cara membaca huruf kanji beserta artinya, peneliti membuat sebuah aplikasi android yang dapat mendeteksi huruf kanji bahasa Jepang menggunakan kamera *smartphone* dan menampilkan arti dan cara membacanya pada layar *smartphone*. Dalam membuat aplikasi android ini menggunakan pemrograman Java dan XML pada android studio. Kemudian pada android studio menambahkan *library* baru *tesseract OCR engine* dan *yandex translate API*. *Tesseract OCR engine* berfungsi sebagai pendeteksi huruf kanji menggunakan kamera *smartphone* dan *yandex translate API* berfungsi sebagai penerjemah bahasa Jepang ke bahasa Indonesia. Untuk memperoleh cara membaca huruf kanji Jepang, peneliti membuat database baru menggunakan *SQLite* yang digunakan untuk mengubah dari huruf kanji ke dalam alfabet. Pengujian aplikasi menggunakan metode pengujian *black box* dan mendapatkan tingkat akurasi terhadap *tesseract OCR engine* sebesar 87.5%.

**Kata kunci:** Tesseract OCR Engine, Bahasa Jepang, dan Java.

### 1. Pendahuluan

Bahasa Jepang banyak digemari belakangan ini disebabkan adanya pengaruh dari budaya Jepang melalui anime (kartun Jepang), *cosplay*, dan film yang tersebar luas di Indonesia melalui berbagai media, salah satunya internet [1]. Setiap orang yang mulai tertarik dan menyukai hal-hal yang berhubungan dengan Jepang, pasti mereka akan ikut mempelajari bahasa dan budaya Jepang. Pada penelitian sebelumnya yang dijadikan referensi pada penelitian tugas akhir ini salah satunya Penelitian tentang perancangan sistem untuk membuat aplikasi android yang bisa menerjemahkan dari bahasa Inggris menjadi bahasa Indonesia dengan memanfaatkan *Tesseract OCR Engine* bersifat *open source* yang sekarang dikembangkan oleh Google. Android studio yang digunakan untuk membuat aplikasi pada penelitian ini dengan menerapkan pemrograman Java sebagai bahasa pemrograman pengkodean dan XML sebagai pembuat layout pada aplikasi. Aplikasi ini menggunakan metode *white box* sebagai pengujian aplikasi penerjemah OCR dan memiliki presentase akurasi mencapai 97.5 % [2].

Berbeda dengan penelitian sebelumnya, peneliti saat ini ingin membuat aplikasi android dengan maksud melakukan pengembangan dalam pengujian terhadap *Tesseract OCR Engine* untuk mendeteksi teks bahasa yang berbeda, yaitu bahasa Jepang. Software android studio dengan menggunakan bahasa pemrograman java dan XML untuk membuat aplikasi android pada penelitian ini. Untuk metode pengujian yang digunakan pada aplikasi adalah menggunakan metode *black box testing*.

## 2. Tinjauan Pustaka

### 2.1 Bahasa Jepang

Bahasa Jepang memiliki 3 huruf utama, yaitu huruf hiragana, katakana, kanji [3]. Hiragana merupakan huruf bahasa Jepang asli yang dibuat sendiri oleh orang Jepang. Huruf hiragana memiliki fungsi sebagai kosa-kata asli Bahasa Jepang yang bukan dari kata serapan atau mengambil dari bahasa asing. Katakana merupakan salah satu jenis huruf bahasa Jepang yang digunakan untuk menulis kata serapan dari bahasa asing dan juga menulis nama orang yang tidak menggunakan nama orang Jepang [4]. Kanji merupakan salah satu jenis huruf bahasa Jepang yang paling sering digunakan oleh orang Jepang. Berbeda dengan jenis huruf sebelumnya seperti hiragana, hiragana digunakan oleh anak-anak di sekolah negara Jepang untuk belajar cara membaca huruf kanji. Huruf kanji bukanlah huruf asli yang dibuat oleh orang Jepang, melainkan huruf kanji diambil dari huruf bahasa Cina yang kemudian diterapkan dan dikembangkan menjadi huruf kanji [5].

### 2.2 Tesseract OCR Engine

*Optical Character Recognition (OCR)* merupakan *software* aplikasi yang digunakan untuk mendeteksi sebuah teks, angka, maupun sebuah pola karakter dari sebuah gambar [6]. Sedangkan *Tesseract OCR engine* merupakan sebuah mesin *optical character recognition* yang sekarang dikembangkan oleh Google. Setelah hasil dari pendeteksian *Tesseract OCR* pada gambar teks didapatkan kemudian dilakukan perhitungan dengan menguji tingkat keakuratan pada pengenalan karakter atau *Character Accuracy (Cacc)* menggunakan persamaan (1) [2]:

$$\text{acc} = \frac{\text{jumlah teks yang dikenali}}{\text{total semua teks masukan}} \times 100 \% \quad (1)$$

### 2.3 Bahasa Pemrograman Java dan XML

Bahasa pemrograman Java merupakan salah satu bahasa pemrograman yang paling sering digunakan dalam membuat game atau sebuah aplikasi berbasis android. Pemrograman Java juga digunakan sebagai perintah program untuk menjalankan fungsi atribut pada layout XML [7]. XML (*eXtensible Markup Language*) pada pemrograman aplikasi android digunakan untuk membuat layout atau tampilan antarmuka dari aplikasi yang akan dibuat sesuai kebutuhan [8].

### 2.4 Black Box Testing

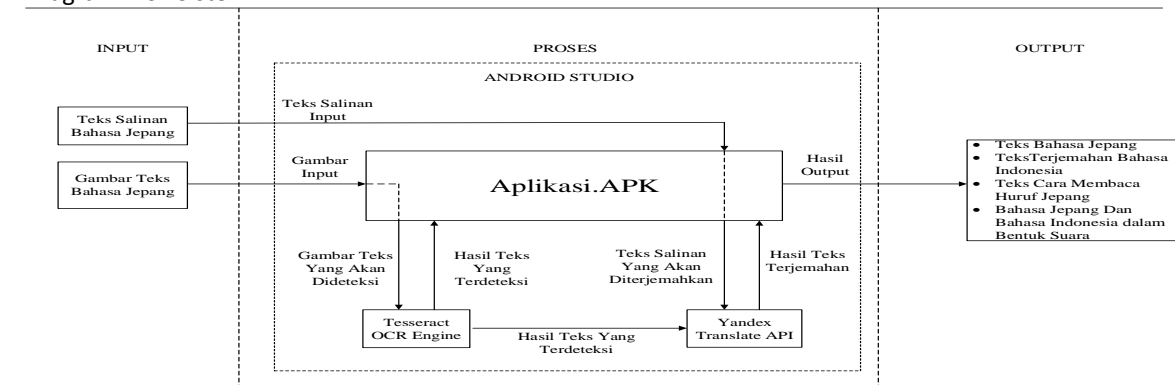
*Black box testing* merupakan salah satu metode yang dapat digunakan untuk menguji suatu perangkat lunak atau sebuah aplikasi yang baru dibuat. Metode *Black box testing* cara pengujiannya adalah dengan memfokuskan pengujian terhadap hasil keluaran yang diperoleh dari sebuah aplikasi yang disesuaikan dengan tujuan awal dari pembuatan aplikasi tersebut [9].

## 3. Perancangan Sistem

### 3.1 Perancangan Sistem Aplikasi

Ada beberapa perancangan sistem yang dibutuhkan dalam membuat aplikasi penerjemah bahasa Jepang ini, antara lain:

#### a. Diagram Blok Sistem

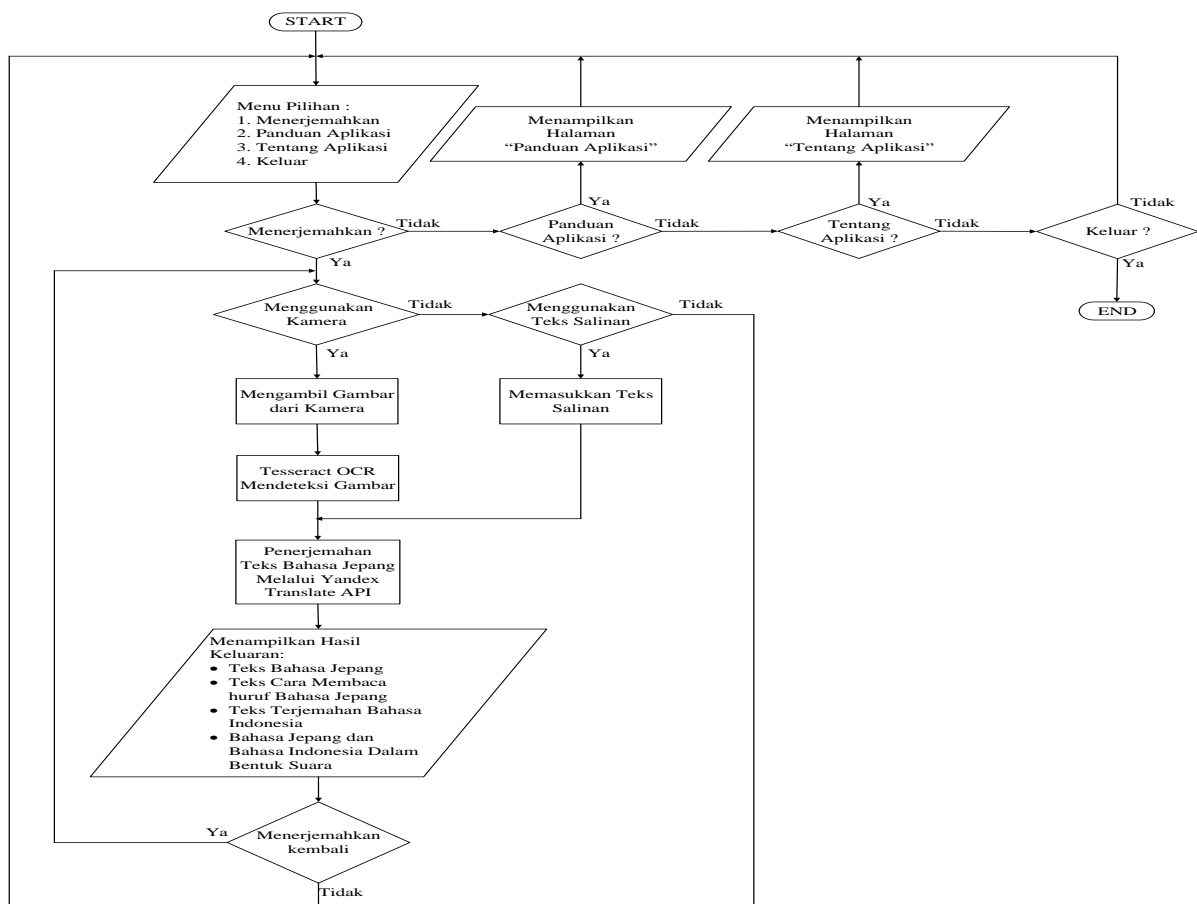


Gambar 1. Diagram Blok Sistem

Gambar 1 menunjukkan gambaran diagram blok sistem dengan 2 cara inputan yang dapat digunakan pada aplikasi, yaitu dengan gambar teks dan teks salinan. Inputan dari gambar teks menjadi faktor utama pada aplikasi yang akan dibuat. Gambar teks bahasa Jepang yang akan diterjemahkan diambil melalui aplikasi, kemudian masuk ketahapan Tesseract OCR Engine sebagai tahapan proses pendeteksian teks Jepang, setelah teks Jepang terdeteksi, kemudian dilanjutkan ketahapan proses menampung hasil teks yang telah terdeteksi ke dalam aplikasi. Hasil teks yang terdeteksi sebelumnya juga akan dilanjutkan ketahapan proses terjemahan ke dalam *yandex translate API*, kemudian hasil dari terjemahan teks yang terdeteksi akan ditampung kembali ke dalam aplikasi. Setelah mendapat teks yang dideteksi dan hasil terjemahan bahasa Indonesia nya dari gambar teks, maka di dalam proses aplikasi akan Menambahkan dalam bentuk lain hasil dari kedua teks tersebut yaitu ke dalam bentuk suara. Setelah itu hasil dari teks dan suaranya akan dikeluarkan dilayar *smartphone*. Untuk inputan dengan teks salinan prosesnya hampir sam dengan inputan dengan gambar teks, yang berbeda hanyalah tidak adanya proses tahapan tesseract OCR untuk pendeteksian teks.

#### b. Flowchart Sistem

Flowchart atau diagram alir menunjukkan aliran kerja sistem secara keseluruhan. Diagram alir di bawah ini menggambarkan urutan dari prosedur-prosedur yang ada pada sistem aplikasi yang akan dibuat serta memperlihatkan cara kerja dari aplikasi tersebut.



**Gambar 2.** Flowchart Sistem Keseluruhan Aplikasi

Pada Gambar 2 merupakan flowchart cara kerja aplikasi yang akan dirancang. Aplikasi memiliki tampilan pada halaman menu utama dengan 4 pilihan, yaitu "Menerjemahkan", "Panduan Aplikasi", "Tentang Aplikasi", dan "Keluar". Ketika menggunakan pilihan "Menerjemahkan", maka akan keluar tampilan memulai terjemahan dengan cara memilih menggunakan kamera atau dengan salinan teks secara langsung. Saat menggunakan kamera, maka akan membuka kamera, setelah gambar teks telah diambil dari kamera tersebut selanjutnya akan masuk ke dalam proses pendeteksian teks menggunakan tesseract OCR untuk mendapatkan teks bahasa Jepang yang ada pada gambar. Teks yang telah terdeteksi akan ditampilkan di kolom teks. Kemudian hasil dari teks yang didapatkan dari gambar akan masuk ke dalam proses terjemahan dari *yandex translate API*. Setelah hasil dari terjemahan dalam bahasa Indonesia didapatkan, maka akan ditampilkan pada kolom teks. Setelah semua proses selesai berjalan, maka akan menampilkan hasil akhir keseluruhan berupa teks bahasa Jepang dari pendeteksi gambar, teks cara

membaca huruf bahasa jepangnya, teks terjemahan bahasa Indonesia, dan bahasa Jepang - bahasa Indonesia dalam bentuk suara. Sedangkan saat menggunakan teks salinan proses dari penerjemahan sama dengan menggunakan teks, yang berbeda hanyalah tidak adanya proses tesseract OCR mendeteksi teks.

#### 4. Hasil Penelitian dan Pembahasan

Pada pengujian sistem aplikasi yang telah dilakukan merupakan pengujian terhadap hasil keluaran dari aplikasi yang selesai dibuat untuk mengetahui hasil keluaran nya tersebut sesuai dengan yang telah dirancang sebelumnya.

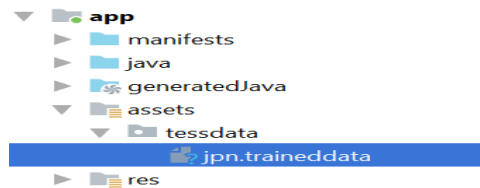
##### 4.1 Tahap Implementasi Sistem Aplikasi

Pada tahap implemantasi sistem aplikasi dilakukan berdasarkan hasil dari perancangan sistem yang kemudian di implementasikan. Selain android studio yang digunakan sebagai software utama dalam pembuatan aplikasi, ada beberapa software lain yang dijadikan *library* pada android studio yaitu, *Tesseract OCR Engine*. Tesseract OCR berfungsi untuk pengenalan atau pendeteksi karakter huruf jepang pada sebuah gambar. Untuk tahap pertama, menjadikan Tesseract OCR menjadi *library* pada lingkungan android studio membutuhkan script program tambahan pada app module build grandle.

```
dependencies {
    implementation 'com.rmtheis:tess-two:9.0.0'
}
```

Gambar 3. Memasukkan Tesseract OCR Engine Pada Android Studio

Pada Gambar 3 merupakan script program awal pada app module build grandle android studio yang berguna untuk menambahkan *library Tesseract OCR engine*. Setelah melakukan penambahan perintah script program, kemudian klik "Sync" untuk mengunduh dan menghubungkan secara otomatis komponen-komponen yang dibutuhkan antara Tesseract OCR dan android studio.



Gambar 4. Menambahkan Data Pengenalan Huruf Jepang

Gambar 4 merupakan tahapan lanjutan untuk menambahkan *library Tesseract OCR Engine*. Setelah proses sinkronisasi Tesseract OCR pada Gambar 3 selesai berjalan, kemudian membuat assest folder pada app dan menambahkan resource directory baru pada assest folder yang didalam nya telah dimasukkan traineddata bertipe bahasa jepang.

<pre>import com.googlecode.tesseract.android.TessBaseAPI;  private TessBaseAPI mTess; String datapath = "";  String language = "jpn"; datapath = getFilesDir() + "/tesseract/"; mTess = new TessBaseAPI(); checkFile(new File(datapath + "tessdata/")); mTess.init(datapath, language); //proses perintah pada Tesseract OCR private void checkFile(File dir) {     if (!dir.exists() &amp;&amp; dir.mkdirs()) {         copyFiles();     }     if (dir.exists()) {         String datafilepath = datapath + "/tessdata/jpn.traineddata";         File datafile = new File(datafilepath);          if (!datafile.exists()) {             copyFiles();         }     } } //proses perintah pada Tesseract OCR private void copyFiles() {     try {         String filepath = datapath + "/tessdata/jpn.traineddata";         AssetManager assetManager = getAssets();          InputStream instream = assetManager.open("tessdata/jpn.traineddata");         OutputStream outstream = new FileOutputStream(filepath);          byte[] buffer = new byte[1024];         int read;         while ((read = instream.read(buffer)) != -1) {             outstream.write(buffer, 0, read);         }     } }</pre>	<pre>outstream.flush(); outstream.close(); instream.close();  File file = new File(filepath); if (!file.exists()) {     throw new FileNotFoundException(); } } catch (FileNotFoundException e) {     e.printStackTrace(); } catch (IOException e) {     e.printStackTrace(); } }  //hasil dari pengambilan gambar akan dijadikan satuan pixel gambar kemudian masuk kedalam proses //pendeteksian pada Tesseract OCR @Override protected void onActivityResult(int requestCode, int resultCode, Intent data) {     if (requestCode == REQUEST_TAKE_PHOTO         &amp;&amp; resultCode == Activity.RESULT_OK) {         Bundle extra = data.getExtras();         Bitmap photo = (Bitmap) extra.get("data");         fotoKamera.setImageBitmap(photo);         String OCRresult = null;         mTess.setImage(photo);         OCRresult = mTess.getUTF8Text();         EditText OCRTextView = (EditText) findViewById(R.id.inputTeks);         OCRTextView.setText(OCRresult);     } }</pre>
a	b

**Gambar 5.** Script Program *Tesseract OCR Engine*. a) Gambar Potongan Script Program OCR pertama, b) Gambar Script Program OCR Kedua

Gambar 5 merupakan tahapan terakhir untuk menggunakan tesseract OCR, yaitu dengan cara membuat pemrograman java pada *class activity* yang membutuhkan penggunaan tesseract OCR di dalamnya yang dimulai dari *import library* tesseract OCR, inisialisasi awal, pemanggilan Tesseract OCR, merubah gambar dalam bentuk *bitmap*, dan mendeteksi gambar. Setelah proses dari tahap menambahkan *library* tesseract OCR selesai, selanjutnya tahap kedua dengan menghubungkan yandex translate API pada android studio yang nantinya akan digunakan sebagai penerjemah dari aplikasi. Untuk menambahkan yandex translate API ke dalam lingkungan android studio membutuhkan script tambahan class *AsyncTask*.

```
//memanggil URL dan memasukkan kode key dari yandex translate API
String yandexKey = "trnsl.1.1.20190216T082816Z.caa5a6026b26fa78.ca341cc5a55cfd7388411e62db53bef6fead330c";
String yandexUrl = "https://translate.yandex.net/api/v1.5/tr.json/translate?key=" + yandexKey
    + "&text=" + textToBeTranslated + "&lang=" + languagePair;
URL yandexTranslateURL = new URL(yandexUrl);

// menset hubungan HTTP, Input Stream, and Buffered Reader
URLConnection httpURLConnection = (URLConnection) yandexTranslateURL.openConnection();
InputStream inputStream = httpURLConnection.getInputStream();
BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(inputStream));
StringBuilder sb = new StringBuilder();
String buf = bufferedReader.readLine();
while (buf != null) {
    sb.append(buf);
    buf = bufferedReader.readLine();
}
String data = sb.toString();
JSONObject jsonObj = new JSONObject(data);
String string = jsonObj.getString("text");
resultString = string.substring(string.indexOf('[')+2, string.indexOf(']')-1);
return resultString;
```

**Gambar 6.** Menghubungkan Yandex Translate API Pada Android Studio

Pada Gambar 6 merupakan potongan script program untuk menghubungkan yandex translate API dari android studio dengan menghubungkan URL dan *code key* yang didapat pada developer yandex.

**4.2 Hasil Pengujian Aplikasi**

Pengujian yang dilakukan pada aplikasi ini menggunakan metode pengujian *black box* dengan media kertas A4 yang tercetak dari printer dengan font MS Gothic sebagai media uji aplikasi. Pada Percobaan Tabel 1, untuk menguji pendeteksian OCR pada aplikasi dengan menggunakan kamera tanpa *flash* dan *flash* dari segi ukuran besarnya huruf teks jepang.

**Tabel 1.** Percobaan Pada Pendeteksian OCR *Font Size 12 dan 14*

Input Data	Teks Font 12 Tanpa Flash	Teks Font 12 dengan Flash	Teks Font 14 Tanpa Flash	Teks Font 14 dengan Flash
火	火	火	火	火
橋	Tidak sesuai	橋	橋	橋
鶏	Tidak sesuai	Tidak sesuai	鶏	鶏
靴	靴	Tidak sesuai	靴	靴
扉	扉	Tidak sesuai	扉	扉
真似る	真似る	Tidak sesuai	真似る	真似る
寝る	寝る	Tidak sesuai	寝る	寝る
立つ	立つ	立つ	立つ	立つ
歌う	Tidak sesuai	歌う	歌う	歌う
飛ぶ	飛ぶ	飛ぶ	飛ぶ	飛ぶ
美味しい	美味しい	美味しい	美味しい	美味しい
若い	Tidak sesuai	Tidak sesuai	若い	若い
甘い	甘い	甘い	甘い	甘い
可愛い	Tidak sesuai	Tidak sesuai	可愛い	可愛い
楽しい	Tidak sesuai	Tidak sesuai	楽しい	楽しい
大切	大切	大切	大切	大切
丁寧	Tidak sesuai	Tidak sesuai	Tidak sesuai	丁寧

有名	有名	有名	有名	有名
親切	親切	親切	親切	親切
安全	安全	安全	安全	安全
誕生日	誕生日	誕生日	誕生日	誕生日
機械語	機械語	Tidak sesuai	機械語	機械語
土曜日	土曜日	Tidak sesuai	土曜日	土曜日
自動車	自動車	自動車	自動車	自動車
電気工学	Tidak sesuai	電気工学	電気工学	電気工学
私はインドネシア人です	私はインドネシア人です	Tidak sesuai	私はインドネシア人です	私はインドネシア人です
学校で友達と会います	学校で友達と会います	Tidak sesuai	学校で友達と会います	学校で友達と会います
桜の花をみたいです	桜の花をみたいです	桜の花をみたいです	桜の花をみたいです	桜の花をみたいです
日本語で話さない	Tidak sesuai	Tidak sesuai	日本語で話さない	日本語で話さない
ペンで書きます	Tidak sesuai	Tidak sesuai	Tidak sesuai	Tidak sesuai

Pada tabel 1 menunjukkan hasil perbandingan dari pendeteksian OCR pada aplikasi dengan ukuran teks 12 dan 14 menggunakan kamera tanpa flash dan menggunakan flash. Dari percobaan sebanyak 30 data, pendeteksian menggunakan kamera tanpa flash untuk *font* 12 berhasil mendeteksi 20 data, sedangkan dengan menggunakan *flash* kamera 15 data. Untuk ukuran *font* 14 menggunakan kamera tanpa flash berhasil mendeteksi 28 data, sedangkan dengan menggunakan *flash* kamera 29 data.

Pada Tabel 2 menunjukkan hasil perbandingan dari pendeteksian OCR pada aplikasi dengan ukuran teks 16 dan 18 menggunakan kamera tanpa flash dan menggunakan flash. Dari percobaan sebanyak 30 data, pendeteksian menggunakan kamera tanpa flash untuk *font* 16 berhasil mendeteksi 28 data, sedangkan dengan menggunakan *flash* kamera 30 data. Untuk ukuran *font* 18 menggunakan kamera tanpa *flash* maupun menggunakan *flash* kamera berhasil mendeteksi 30 data. Untuk percobaan yang dilakukan dengan menguji dari segi ukuran huruf dari 12, 14, 16, dan 18 dengan tanpa menggunakan *flash* kamera dan menggunakan *flash* kamera dapat disimpulkan bahwa ukuran dan pencahayaan dapat mempengaruhi dari hasil pendeteksian, semakin besar ukuran teks jepang yang dideteksi maka semakin tinggi keberhasilan dari hasil pendeteksian yang akan didapat.

Dengan adanya percobaan pada Tabel 1 dan Tabel 2 didapatkan keakurasian menggunakan tesseract OCR pada aplikasi ini yaitu dengan menghitung seluruh data yang digunakan untuk melakukan percobaan pada OCR yang terdiri dari 30 input teks data, untuk jumlah total percobaan sebanyak 240 data, yang berhasil terdeteksi sesuai dengan gambar teks sebanyak 210 data dengan kesalahan pendeteksian sebanyak 30 data. Kemudian dari hasil data yang diperoleh dilakukan perhitungan tingkat akurasi dari pengenalan karakter (Cacc) seperti pada persamaan (1) dan mendapatkan hasil:

$$\text{acc} = \frac{210}{240} = 0.875 \times 100 \% = 87.5 \%$$

Jadi, akurasi tingkat pendeteksian pada aplikasi ini sebesar 87.5 % yang didapatkan dari perhitungan diatas. Pendeteksian OCR pada aplikasi gagal mendeteksi sesuai dengan teks yang ada pada gambar sebesar 12.5 %.

Pada tabel 3 menunjukkan hasil percobaan perbandingan menerjemahkan pada aplikasi dengan menggunakan kamus dan terjemahan aplikasi. Pada percobaan tabel 3 dari 30 data teks bahasa jepang yang diterjemahkan, aplikasi dapat menerjemahkan sesuai dengan kamus sebanyak 23 data. Dilihat dari hasil dari tabel 4.4 pada percobaan ini dapat disimpulkan bahwa untuk menerjemahkan kosa-kata akurasi keberhasilannya tinggi, sedangkan untuk menerjemahkan dalam bentuk kalimat persentase keberhasilannya rendah. Untuk semua teks terjemahan bahasa Indonesia berhasil mengeluarkan dalam bentuk suara sesuai ejaan teks terjemahannya.

Tabel 2. Percobaan Pada Pendeteksian OCR Font Size 16 dan 18

Input Data	Teks Font 16 Tanpa Flash	Teks Font 16 dengan Flash	Teks Font 18 Tanpa Flash	Teks Font 18 dengan Flash
火	火	火	火	火
橋	橋	橋	橋	橋
鶏	鶏	鶏	鶏	鶏
靴	靴	靴	靴	靴
扉	扉	扉	扉	扉
真似る	真似る	真似る	真似る	真似る
寝る	寝る	寝る	寝る	寝る
立つ	立つ	立つ	立つ	立つ
歌う	歌う	歌う	歌う	歌う
飛ぶ	飛ぶ	飛ぶ	飛ぶ	飛ぶ
美味しい	美味しい	美味しい	美味しい	美味しい
若い	若い	若い	若い	若い
甘い	甘い	甘い	甘い	甘い
可愛い	可愛い	可愛い	可愛い	可愛い
楽しい	楽しい	楽しい	楽しい	楽しい
大切	大切	大切	大切	大切
丁寧	Tidak sesuai	丁寧	丁寧	丁寧
有名	有名	有名	有名	有名
親切	親切	親切	親切	親切
安全	安全	安全	安全	安全
誕生日	誕生日	誕生日	誕生日	誕生日
機械語	機械語	機械語	機械語	機械語
土曜日	土曜日	土曜日	土曜日	土曜日
自動車	自動車	自動車	自動車	自動車
電気工学	電気工学	電気工学	電気工学	電気工学
私はインドネシ ア人です	私はインドネシア 人です	私はインドネ シア人です	私はインドネシ ア人です	私はインドネ シア人です
学校で友達と会 います	学校で友達と会い ます	学校で友達と 会います	学校で友達と会 います	学校で友達と 会います
桜の花をみたい です	桜の花をみたいで す	桜の花をみた いです	桜の花をみたい です	桜の花をみた いです
日本語で話しな さい	日本語で話しなさ い	日本語で話し なさい	日本語で話しな さい	日本語で話し なさい
ペンで書きます	Tidak sesuai	ペンで書きま す	ペンで書きます	ペンで書きま す

Tabel 3. Percobaan Penerjemahan Pada Aplikasi

Input Data	Terjemahan Kamus[10]	Terjemahan Aplikasi	Penyesuaian	Bentuk Suara
火	Api	Api	Sesuai	Berhasil
橋	Jembatan	Jembatan	Sesuai	Berhasil
鶏	Ayam	Ayam	Sesuai	Berhasil
靴	Tas	Tas	Sesuai	Berhasil
扉	Pintu	Pintu	Sesuai	Berhasil
真似る	Meniru	Meniru yang	Tidak Sesuai	Berhasil
寝る	Tidur	Tidur	Sesuai	Berhasil
立つ	Berdiri	Berdiri	Sesuai	Berhasil
歌う	Menyanyi	Bernyanyi	Sesuai	Berhasil
飛ぶ	Terbang	Terbang	Sesuai	Berhasil
美味しい	Enak	Lezat	Sesuai	Berhasil

若い	Muda	Muda	Sesuai	Berhasil
甘い	Manis	Manis	Sesuai	Berhasil
可愛い	lucu	lucu	Sesuai	Berhasil
楽しい	Menyenangkan	Menyenangkan	Sesuai	Berhasil
大切	Penting	Penting	Sesuai	Berhasil
丁寧	Sopan santun, teliti	Sopan	Sesuai	Berhasil
有名	Terkenal	Terkenal	Sesuai	Berhasil
親切	Baik, ramah	Silahkan	Tidak Sesuai	Berhasil
安全	Aman, Selamat	Aman	Sesuai	Berhasil
誕生日	Hari ulang tahun	ulang tahun	Sesuai	Berhasil
機械語	Bahasa mesin	Bahasa mesin	Sesuai	Berhasil
土曜日	Hari Sabtu	Sabtu	Sesuai	Berhasil
自動車	Mobil	Otomotif	Tidak Sesuai	Berhasil
電気工学	Teknik Elektro/Listrik	Teknik Listrik	Sesuai	Berhasil
私はインドネシア人です	Saya orang Indonesia	Aku orang Indonesia	Sesuai	Berhasil
学校で友達と会います	Saya bertemu teman di sekolah	Teman-teman anda di sekolah dan bertemu	Tidak Sesuai	Berhasil
桜の花をみたいです	Ingin melihat bunga sakura	Bunga sakura yang sedang...	Tidak Sesuai	Berhasil
日本語で話さない	Berbicaralah menggunakan bahasa jepang	Bicara Bahasa Inggris	Tidak Sesuai	Berhasil
ペンで書きます	Menulis menggunakan pensil	pen	Tidak Sesuai	Berhasil

Tabel 4. Percobaan Perubahan Huruf Kanji dalam Bentuk Alfabet

Input Data	Kamus [10]	Aplikasi	Penyesuaian	Bentuk Suara
火	Hi	Hi	Sesuai	Berhasil
橋	Hashi	Hashi	Sesuai	Berhasil
鶏	Niwatori	Niwatori	Sesuai	Berhasil
靴	Kaban	Kaban	Sesuai	Berhasil
扉	Tobira	Tobira	Sesuai	Berhasil
真似る	Maneru	Maneru	Sesuai	Berhasil
寝る	Neru	Neru	Sesuai	Berhasil
立つ	Tatsu	Tatsu	Sesuai	Berhasil
歌う	Utau	Utau	Sesuai	Berhasil
飛ぶ	Tobu	Tobu	Sesuai	Berhasil
美味しい	Oishii	Oishii	Sesuai	Berhasil
若い	Wakai	Wakai	Sesuai	Berhasil
甘い	Amai	Amai	Sesuai	Berhasil
可愛い	kawaii	kawaii	Sesuai	Berhasil
楽しい	Tanoshi	Tanoshi	Sesuai	Berhasil
大切	Taisetsu	Taisetsu	Sesuai	Berhasil
丁寧	Teinei	Teinei	Sesuai	Berhasil
有名	Yuumei	Yuumei	Sesuai	Berhasil
親切	Shinsetsu	Shinsetsu	Sesuai	Berhasil
安全	Anzen	Anzen	Sesuai	Berhasil
誕生日	Tanjoubi	Tanjoubi	Sesuai	Berhasil
機械語	Kikaigo	Kikaigo	Sesuai	Berhasil
土曜日	Doyoubi	Doyoubi	Sesuai	Berhasil
自動車	Jidousha	Jidousha	Sesuai	Berhasil
電気工学	Denkikougaku	Denkikougaku	Sesuai	Berhasil

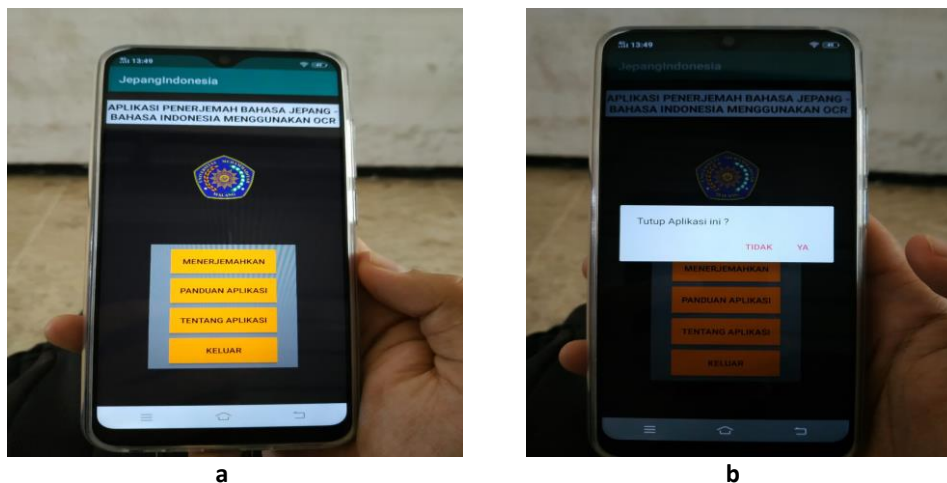


私はインドネシア人です	Watashi wa indonesia jin desu	-	Tidak Sesuai	Berhasil
学校で友達と会います	Gakkou de tomodachi to aimasu	-	Tidak Sesuai	Berhasil
桜の花をみたいです	Sakura no hana wo mitai desu	-	Tidak Sesuai	Berhasil
日本語で話さない	Nihonggo de hanashinasai	-	Tidak Sesuai	Berhasil
ペンで書きます	Pen de kakimashita	-	Tidak Sesuai	Berhasil

Pada tabel 4 menunjukkan hasil percobaan perbandingan dari kamus dan aplikasi terhadap hasil perubahan dari teks jepang ke dalam bentuk huruf alfabet yang berguna untuk membaca teks jepang tersebut. Perubahan dari teks jepang ke dalam bentuk huruf alphabet tidak bisa digunakan untuk sebuah kalimat, tetapi hanya bisa digunakan untuk satu kata dasar seperti yang terlihat pada tabel 4 Untuk semua teks bahasa jepang berhasil mengeluarkan dalam bentuk suara sesuai ejaan teks jepang nya.

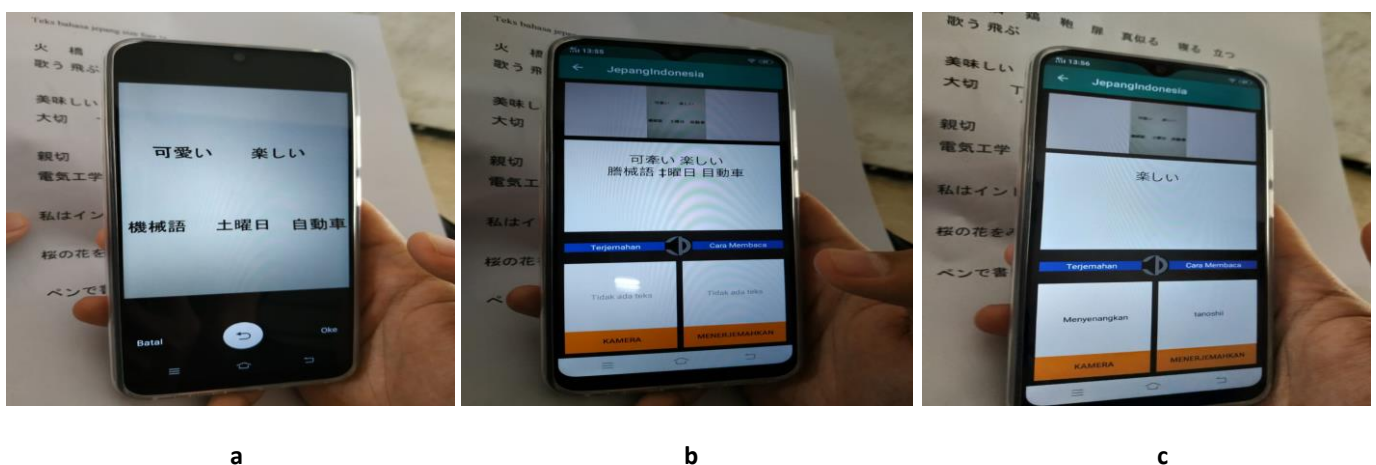
### 4.3 Hasil Pengujian Aplikasi Pada *Smartphone*

Pengujian Aplikasi pada *smartphone* secara langsung bertujuan untuk mengetahui apakah aplikasi dapat berjalan dengan baik saat digunakan dan sesuai dengan perancangan pembuatan aplikasi.

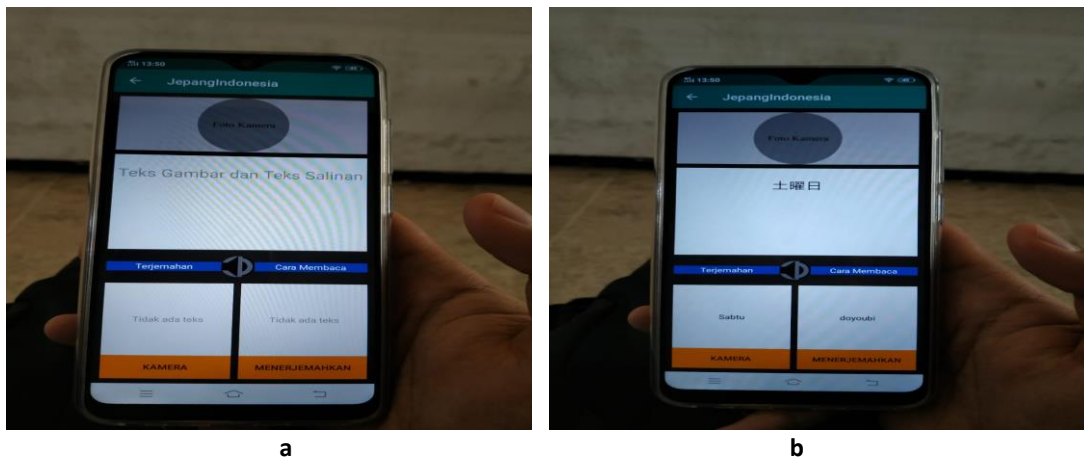


**Gambar 5.** Tampilan Halaman Utama Aplikasi. a) Gambar Halaman Utama, b) Gambar Keluar Aplikasi

Pada gambar 5 menampilkan tampilan pada layar *smartphone* pengguna pada saat aplikasi dibuka dan tampilan saat akan menutup aplikasi. Pada Gambar 6 menampilkan saat pengambilan gambar teks, teks pada gambar telah terdeteksi dan menampilkan hasil teks yang terdeteksi pada gambar di kolom teks, kemudian menampilkan hasil dari terjemahan dan perubahan huruf alfabetnya pada kolom teks masing-masing.

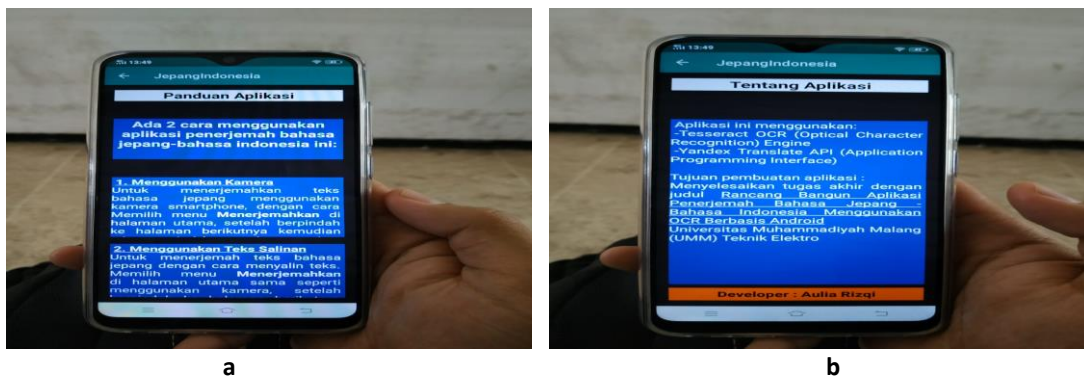


**Gambar 6.** Tampilan Halaman Menerjemahkan Menggunakan Kamera. a) Gambar Pengambilan Teks dengan Kamera, b) Gambar Mendeteksi Teks Bahasa Jepang, c) Gambar Hasil Menerjemahkan



**Gambar 7.** Tampilan Halaman Menerjemahkan Menggunakan Teks Salinan. a) Gambar Tampilan Sebelum Menyalin Teks Jepang, b) Gambar Hasil Menerjemahkan Teks Salinan

Pada gambar 7 menampilkan tampilan sebelum menyalin teks bahasa jepang kemudian tampilan dari hasil terjemahan menggunakan teks bahasa jepang yang disalin dan diletakkan pada kolom teks aplikasi.



**Gambar 8.** Tampilan Halaman Panduan dan Tentang Aplikasi di Layar *Smartphone*. a) Gambar Tampilan Panduan Aplikasi, b) Gambar Tampilan Panduan Aplikasi

Pada gambar 8 menampilkan tampilan panduan cara untuk menggunakan aplikasi untuk para pengguna aplikasi dan tampilan berisikan mengenai tentang aplikasi penerjemah bahasa jepang – bahasa Indonesia.

## 5. Kesimpulan

Berdasarkan hasil uji yang telah dilakukan pada aplikasi, besar kecilnya ukuran teks huruf bahasa jepang dan pencahayaan dilingkungan sekitar dapat mempengaruhi hasil dari pendeteksian OCR pada sebuah gambar teks. Tingkat akurasi keberhasilan OCR pada aplikasi dalam mendeteksi teks pada gambar sebesar 87.5 %.

Persentase keberhasilan untuk menerjemahkan kosa-kata bahasa jepang lebih tinggi dari pada menerjemahkan sebuah kalimat bahasa jepang dan untuk hasil keluaran dalam bentuk suara akan dapat didengarkan apabila teks terjemahan maupun alfabet dari teks bahasa jepang telah diperoleh.

Untuk jarak pengambilan gambar teks untuk mendapatkan pendeteksian yang ideal pada jarak berkisar minimal 6 cm dan maksimal 10 cm dari *smartphone*.

## Referensi

- [1]. Setiawan, A. Suryaningrum, K.M. 2017. Optical Character Recognition Jepang Menggunakan Matriks Populasi Pixel dan L1-Metric. Jurnal Ilmiah Teknologi Informasi Terapan. Volume IV, No 1, 15 Desember. Universitas Widyatama.
- [2]. Utami, A.E., Nurhayati, O.D., & Martono, K.T. 2016. Aplikasi Penerjemah Bahasa Inggris – Indonesia dengan Optical Character Recognition Berbasis Android. Jurnal Teknologi dan Sistem Komputer. Vol.4, No.1, Januari 2016 : 167-177. Universitas Diponegoro.

- [3]. [Anonim]. 2014. Minna no Nihonggo I. Surabaya. International Multicultural (I'Mc) Center Press (Co.publisher Indonesia Edition).
- [4]. Haryanto, R. T., Mahardityawarman, R., Kusnaedi, & Priyanggodo, D. Y. 2016. Pengenalan Tulisan Tangan Karakter Jepang Menggunakan Library Tesseract Pada Android. In S. Widyarto (Ed.), Proceeding of the 2nd Informatics Conference 2016 (ICF2016) (pp. 51-53). Jakarta: Universitas Budi Luhur.
- [5]. Renariah, Dra., M.Hum. 2012. Bahasa Jepang dan Karakteristiknya. Jurnal Sastra Jepang Fakultas Sastra. vol 1 No.2 edisi february 2012. Universitas Kristen Maranatha.
- [6]. Galih A.R.S., Erik, & Hakim, M. L. 2014. Penerapan Teknik OCR (Optical Character Recognition) pada Aplikasi Terjemahan Kitab Fiqih Safinah An-Naja Menggunakan ReadDRIS. Seminar Nasional Infomatika (pp. 60-69). Yogyakarta: UPN "Veteran".
- [7]. Sallaby, A.F., Utami, F.H., Arliando, Y. 2015. Aplikasi Widget Berbasis Java. Jurnal Media Infotama Vol. 11 No. 2. September 2015. Universitas Dehasen Bengkulu.
- [8]. Noprianto, 2004, Mengenal XML. Info Linux 12/2004.
- [9]. Sari, E.D. 2013. Pengembangan Sistem Pengarsipan Surat Di Bidang Sumber Daya Manusia Dan Komunikasi Hukum Administrasi Pada Pt Pln (Persero) Wilayah Lampung. Skripsi. Lampung. Jurusan Ilmu Komputer Fakultas Matematika Dan Ilmu Pengetahuan Alam. Universitas Lampung.
- [10]. Shiang, T. T. 2014. Kiat Sukses Mudah & Praktis Mencapai N5. Jakarta. Gakushudo.