

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

<sup>1</sup>Aulia Rizqi, <sup>2</sup>Izzan Silmi Aziz, <sup>3</sup>Nur Kasan, <sup>4</sup>Widianto

<sup>1,2,3,4</sup>Fakultas Teknik, Jurusan Teknik Elektro, Universitas Muhammadiyah Malang

<sup>1</sup>Auliarizqi1010@gmail.com

**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.

**Abstrak**—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.

## I. 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*.

## II. Metode Penelitian

### A. 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].

**B. Tesseract OCR Engine**

Pada bagian ini, penulis menampilkan metode yang digunakan, termasuk di dalamnya waktu dan lokasi penelitian jika diperlukan. *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 keakurasian pada pengenalan karakter atau *Character Accuracy (Cacc)* menggunakan persamaan (1) [2]:

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

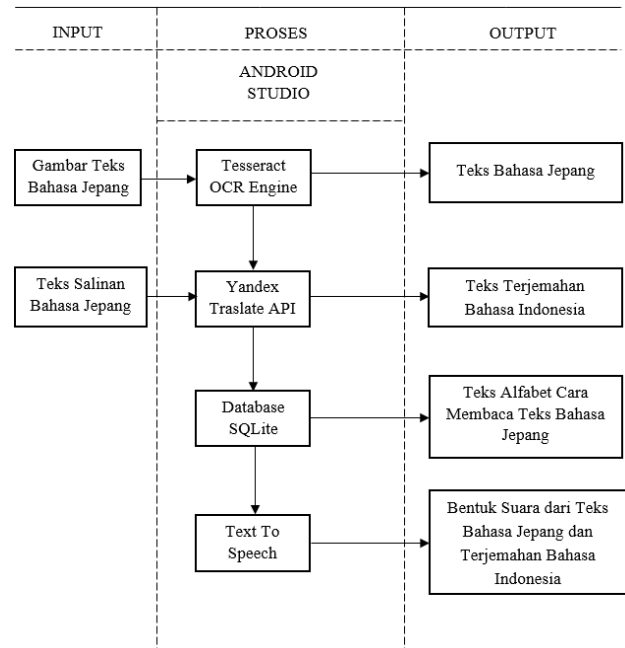
**C. Black Box Testing**

*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].

**D. 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].

**E. Diagram Blok Sistem**



Gambar 1. Diagram Blok Sistem

Gambar 1 merupakan gambaran diagram blok sistem yang memiliki 3 bagian, yaitu input, proses, dan output. Bagian pertama input, input terdiri dari 2 cara yang pertama gambar teks dan yang kedua teks salinan. Inputan yang menggunakan Gambar teks merupakan inputan dengan cara mengambil gambar yang memiliki teks huruf bahasa Jepang untuk diterjemahkan, sedangkan teks salinan merupakan inputan dengan cara menyalin teks bahasa Jepang untuk diterjemahkan. Bagian kedua proses, seluruh proses pembuatan aplikasi ini menggunakan *software* android studio yang terdiri dari *tesseract OCR engine*, *yandex translate API*, *database SQLite*, dan *text to speech*. *Tesseract OCR engine* berfungsi untuk mendeteksi teks bahasa Jepang pada gambar. *Yandex translate API* berfungsi untuk menerjemahkan hasil dari pendeteksian teks bahasa Jepang pada gambar dan menerjemahkan teks salinan. *Database SQLite database* berbentuk teks yang berfungsi untuk mengubah teks bahasa Jepang yang terdeteksi dari gambar maupun dari teks salinan menjadi huruf alfabet. *Text to speech* berfungsi untuk mengubah teks bahasa Jepang dan hasil terjemahan ke dalam bentuk suara. Bagian ketiga output, output merupakan hasil keluaran yang diperoleh dari proses keseluruhan yang terdiri dari teks bahasa Jepang, teks terjemahan bahasa Indonesia, teks alfabet cara membaca teks bahasa Jepang, dan dalam bentuk suara dari teks bahasa Jepang dan terjemahan bahasa Indonesia.

**III. Hasil 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.

3.1 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

No	Input Data	Teks Font 12 Tanpa Flash	Teks Font 12 dengan Flash	Teks Font 14 Tanpa Flash	Teks Font 14 dengan Flash
1	火	火	火	火	火
2	橋	橋	Tidak sesuai	橋	橋
3	鶏	Tidak sesuai	Tidak sesuai	鶏	鶏
4	鞆	Tidak sesuai	鞆	鞆	鞆
5	扉	Tidak sesuai	扉	扉	扉
6	真似る	Tidak sesuai	真似る	真似る	真似る
7	寝る	Tidak sesuai	寝る	寝る	寝る
8	桜の花をみたい 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
9	日本語で話 なさい	Tidak sesuai	Tidak sesuai	日本語で 話 なさい	日本語で 話 なさい
10	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 10 data, pendeteksian menggunakan kamera tanpa flash untuk *font* 12 berhasil mendeteksi 3 data, sedangkan dengan menggunakan *flash* kamera 6 data. Untuk ukuran *font* 14 menggunakan kamera tanpa flash berhasil mendeteksi 9 data, sedangkan dengan menggunakan *flash* kamera 9 data.

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

No	Input Data	Teks Font 16 Tanpa Flash	Teks Font 16 dengan Flash	Teks Font 18 Tanpa Flash	Teks Font 18 dengan Flash
1	火	火	火	火	火

2	橋	橋	橋	橋	橋
3	鶏	鶏	鶏	鶏	鶏
4	鞆	鞆	鞆	鞆	鞆
5	扉	扉	扉	扉	扉
6	真似る	真似る	真似る	真似る	真似る
7	寝る	寝る	寝る	寝る	寝る
8	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
9	日本語で話 なさい	日本語で 話 なさい	日本語で話 なさい	日本語で 話 なさい	日本語で 話 なさい
10	ペンで書 きます	ペンで書 きます	ペンで書 きます	ペンで書 きます	ペンで書 きます

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 10 data, pendeteksian menggunakan kamera tanpa flash untuk *font* 16 berhasil mendeteksi 10 data, sedangkan dengan menggunakan *flash* kamera 10 data. Untuk ukuran *font* 18 menggunakan kamera tanpa *flash* maupun menggunakan *flash* kamera berhasil mendeteksi 10 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.

Tabel 3. Percobaan Penerjemahan Pada Aplikasi

No	Input Data	Terjemahan Kamus[10]	Terjemahan Aplikasi	Penyesuaian	Bentuk Suara
1	火	Api	Api	Sesuai	Berhasil
2	橋	Jembatan	Jembatan	Sesuai	Berhasil
3	鶏	Ayam	Ayam	Sesuai	Berhasil
4	鞆	Tas	Tas	Sesuai	Berhasil
5	扉	Pintu	Pintu	Sesuai	Berhasil
6	真似る	Meniru	Meniru yang	Tidak Sesuai	Berhasil
7	寝る	Tidur	Tidur	Sesuai	Berhasil
8	桜の花を みたい です	Ingin melihat bunga sakura	Bunga sakura yang sedang...	Tidak Sesuai	Berhasil
9	日本語で話 し	Berbicaralah menggunakan bahasa	Bicara Bahasa Inggris	Tidak Sesuai	Berhasil

	なさい	jepang			
10	ペンで書きます	Menulis menggunakan pen	pen	Tidak Sesuai	Berhasil

Pada tabel 3 menunjukkan hasil percobaan perbandingan menerjemahkan pada aplikasi dengan menggunakan kamus dan terjemahan aplikasi. Pada percobaan tabel 3 dari 10 data teks bahasa jepang yang diterjemahkan, aplikasi dapat menerjemahkan sesuai dengan kamus sebanyak 6 data. Dilihat dari hasil dari tabel 3 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.

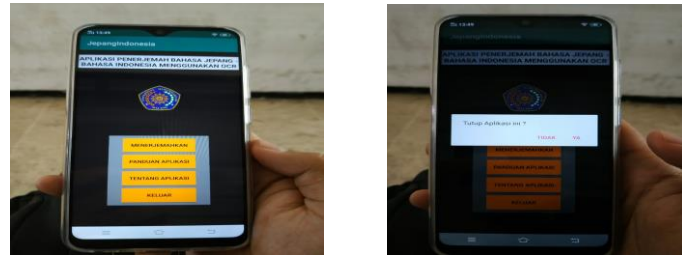
Tabel 4. Percobaan Pada Pendeteksian OCR Font Size 16 dan 18

No	Input Data	Kamus[10]	Aplikasi	Penyesuaian	Bentuk Suara
1	火	Hi	Hi	Sesuai	Berhasil
2	橋	Hashi	Hashi	Sesuai	Berhasil
3	鶏	Niwatori	Niwatori	Sesuai	Berhasil
4	靴	Kaban	Kaban	Sesuai	Berhasil
5	扉	Tobira	Tobira	Sesuai	Berhasil
6	真似る	Maneru	Maneru	Sesuai	Berhasil
7	寝る	Neru	Neru	Sesuai	Berhasil
8	桜の花をみたいです	Sakura no hana wo mitai desu	-	Tidak Sesuai	Berhasil
9	日本語で話さない	Nihonggo de hanashina sai	-	Tidak Sesuai	Berhasil
10	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 yang dibuat menggunakan *database SQLite*. Perubahan dari teks jepang ke dalam bentuk huruf alphabet tidak bisa digunakan untuk sebuah kalimat, tetapi hanya bisa digunakan untuk satu kosakata seperti yang terlihat pada tabel 4 Untuk semua teks bahasa jepang berhasil mengeluarkan dalam bentuk suara sesuai ejaan teks jepang nya.

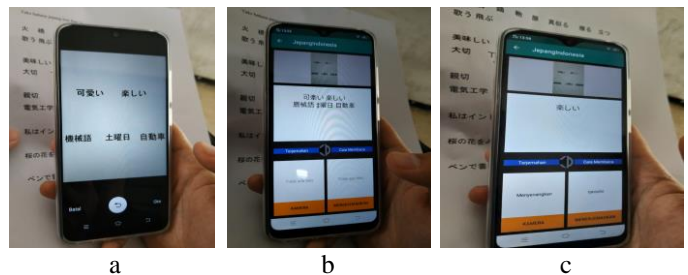
### 3.2 Hasil Pengujian Aplikasi Pada Smartphone

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



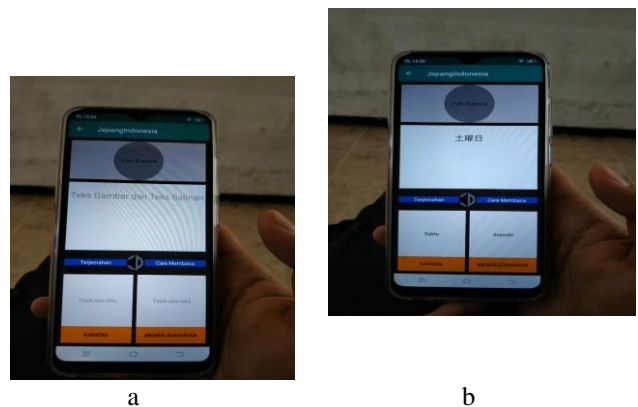
Gambar 2. Tampilan Halaman Utama Aplikasi.  
 a) Gambar Halaman Utama, b) Gambar Keluar Aplikasi

Pada gambar 2 menampilkan tampilan pada layar *smartphone* pengguna pada saat aplikasi dibuka dan tampilan saat akan menutup aplikasi.



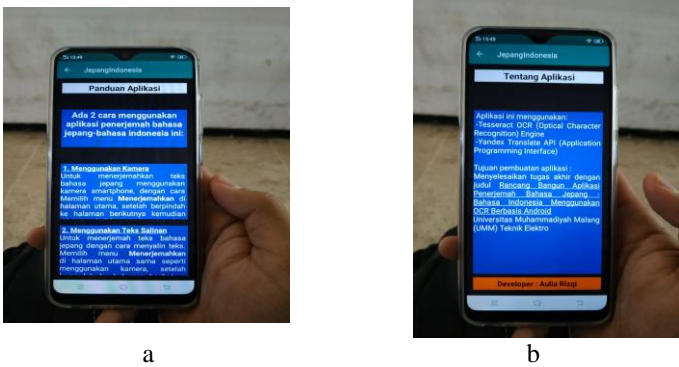
Gambar 3. Tampilan Halaman Menerjemahkan Menggunakan Kamera. a) Gambar Pengambilan Teks dengan Kamera, b) Gambar Mendeteksi Teks Bahasa Jepang, c) Gambar Hasil Menerjemahkan

Pada Gambar 3 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 4. Tampilan Halaman Menerjemahkan Menggunakan Teks Salinan. a) Gambar Tampilan Sebelum Menyalin Teks Jepang, b) Gambar Hasil Menerjemahkan Teks Salinan

Pada gambar 4 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 5. Tampilan Halaman Panduan dan Tentang Aplikasi di Layar Smartphone. a) Gambar Tampilan Panduan Aplikasi, b) Gambar Tampilan Tentang Aplikasi

Pada gambar 5 menampilkan tampilan panduan cara untuk menggunakan aplikasi untuk para pengguna aplikasi dan tampilan berisikan mengenai tentang aplikasi penerjemah bahasa jepang – bahasa Indonesia.

#### IV. Kesimpulan

Kesimpulan dari hasil penelitian dijelaskan pada bagian ini

1. Semakin besar ukuran teks yang dideteksi pada gambar, maka semakin tinggi tingkat akurasi dalam mendeteksi sebuah teks. Penggunaan *flash* pada kamera *smartphone* dapat meningkatkan hasil pendeteksian teks.
2. Hasil ketepatan *yandex translate API* untuk menerjemahkan kosa-kata bahasa jepang lebih tinggi dari pada menerjemahkan sebuah kalimat bahasa jepang.
3. Perubahan teks huruf jepang menjadi huruf alfabet yang dibuat dari *database SQLite* hanya dapat merubah satu persatu kosakata teks bahasa jepang dan untuk hasil keluaran dalam bentuk suara akan dapat didengarkan apabila teks bahasa jepang dan terjemahan bahasa Indonesia didapat pada kolom teks.

#### V. Daftar Pustaka

- [1] A. Setiawan and K. M. Suryaningrum, "Optical Character Recognition Jepang Menggunakan Matriks Populasi Pikel Dan L1-," vol. IV, no. 1, pp. 62–72, 2017.
- [2] A. E. Utami, O. D. Nurhayati, and K. T. Martono, "Aplikasi Penerjemah Bahasa Inggris – Indonesia dengan Optical Character Recognition Berbasis Android," J. Teknol. dan Sist. Komput., vol. 4, no. 1, p. 167, 2016.
- [3] [Anonim]. 2014. Minna no Nihongo I. Surabaya. International Multicultural (I'Mc) Center Press (Co.publisher Indonesia Edition).
- [4] Haryanto, R. T., Mahardiyawarman, 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, "Bahasa Jepang dan karakteristiknya," J. Sastra Jepang Fak. Sastra Univ. Kristen Maranatha, vol. 1, no. 2, pp. 1–16, 2002.
- [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] A. F. Sallaby, F. H. Utami, and Y. Arliando, "Aplikasi widget berbasis java," J. Media Infotama, vol. 11, no. 2, 2015.
- [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.