

PERANCANGAN APLIKASI *TEXT TO SPEECH* DALAM BAHASA INDONESIA MENGUNAKAN *FIREBASE MACHINE LEARNING KIT* BERBASIS ANDROID

Dede Kurniadi^{*1}, Fitri Nuraeni², Indra Trisna Raharja³, Asri Mulyani⁴

^{1,2,3,4}Institut Teknologi Garut, Kabupaten Garut

Email: ¹dede.kurniadi@itg.ac.id, ²fitri.nuraeni@itg.ac.id, ³itraharja@itg.ac.id, ⁴asrimulyani@itg.ac.id

*Penulis Korespondensi

(Naskah masuk: 17 Desember 2021, diterima untuk diterbitkan: 16 Desember 2022)

Abstrak

Aplikasi *text to speech* dapat merubah teks menjadi keluaran suara menggunakan *engine text to speech*, namun teks tersebut harus berupa teks digital agar bisa di render. Sehingga, jika teks berada pada suatu objek maka harus diekstrak terlebih dahulu. *Firebase Machine Learning Kit* menyediakan API *text recognition* untuk membantu proses ekstrak teks. *Firebase Machine Learning Kit* (ML-Kit) juga menyediakan API *language identifier* untuk mendeteksi bahasa pada teks yang dibaca sehingga suara yang dikeluarkan dari teks yang dibaca dapat optimal dengan menggunakan dialek bahasa tertentu. Tujuan dari penelitian ini adalah membangun aplikasi *text to speech* dalam Bahasa Indonesia dengan penerapan *Firebase Machine Learning Kit* berbasis android. Dalam membangun aplikasi ini menggunakan metode *extreme programming* yang tahapannya terdiri dari *planning, design, coding, dan testing*. Hasil dari penelitian ini, berupa aplikasi yang dapat digunakan sebagai alat bantu pembelajaran bahasa asing dan alat digitaisasi teks serta terjemah ke dalam Bahasa Indonesia dan 34 dialek bahasa untuk keluaran suara *text to speech*. Selain itu, pada penelitian ini didapatkan nilai akurasi pengenalan teks dari tulisan tangan dan tulisan mesin, dengan rata-rata persentase akurasi untuk tulisan tangan sebesar 85,25%, sedangkan rata-rata persentase akurasi untuk tulisan mesin sebesar 87,35%. Dengan akurasi yang baik tersebut, maka aplikasi siap untuk dipergunakan sebagai alat bantu dalam proses pembelajaran bahasa asing oleh masyarakat Indonesia.

Kata kunci: *Android, Extreme Programming, Firebase Machine Learning Kit, Text to Speech*

TEXT TO SPEECH APPLICATION DESIGN IN INDONESIAN LANGUAGE USING FIREBASE MACHINE LEARNING KIT BASED ON ANDROID

Abstract

Text to speech applications can convert text into voice output using a text to speech engine, but the text must be digital text in order to render. So, if the text is in an object, it must be extracted first. The Firebase Machine Learning Kit provides a text recognition API to help extract text. The Firebase Machine Learning Kit (ML-Kit) also provides a language identifier API to detect the language in the text being read so that the sound emitted from the text read can be optimized by using a specific language dialect. The purpose of this research is to build a text to speech application in Indonesian with the application of an Android-based Firebase Machine Learning Kit. In building this application using the extreme programming method whose stages consist of planning, design, coding, and testing. The results of this study are in the form of applications that can be used as foreign language learning aids and text digitization tools and translations into Indonesian and 34 language dialects for text to speech voice output. In addition, in this study, the accuracy of text recognition from handwriting and machine writing was obtained, with an average percentage of accuracy for handwriting of 85.25%, while the average percentage of accuracy for machine writing was 87.35%. With good accuracy, the application is ready to be used as a tool in the process of learning foreign languages by the Indonesian people.

Keywords: *Android, Extreme Programming, Firebase Machine Learning Kit, Text to Speech*

1. PENDAHULUAN

Penguasaan bahasa asing menjadi tuntutan tersendiri untuk orang Indonesia di era globalisasi

saat ini. Orang-orang banyak mengikuti kegiatan pelatihan khusus bahasa asing disamping mempelajari bahasa asing dalam kurikulum sekolah atau perguruan tinggi. Namun, kesulitan dalam

penguasaan bahasa asing biasanya adalah menemukan suara baru yang selalu tidak sesuai dengan pelafalan dalam bahasa Indonesia (Husna et al., 2021).

Cara untuk mengetahui pelafalan tulisan dan artinya adalah dengan menuliskan manual di aplikasi penterjemah teks menjadi suara dan aplikasi penterjemah bahasa, atau menanyakannya pada seorang ahli di bidang bahasa asing tersebut. Jika tulisannya sedikit tidak akan menjadi masalah dengan mengetik ulang di aplikasi penterjemah tersebut atau menanyakannya pada seorang ahli, tetapi bagaimana jika tulisannya ada banyak maka akan membutuhkan waktu yang lama.

Hal ini juga terjadi ketika ingin mendokumentasikan tulisan dari buku menjadi tulisan digital yang bisa diolah, kita perlu menulis atau mengetik ulang tulisan bukunya. (Cheriet et al., 2007) mengatakan bahwa *Optical Character Recognition* (OCR) dapat membantu menyelesaikan persoalan ini dalam hal pembacaan dan *extrak* karakter dari pengambilan gambar tulisan tangan atau tulisan mesin. *Machine Learning Kit* (ML Kit) dari *Firestore* dapat digunakan untuk memproses *computing* OCR pada platform android menggunakan *API text recognition* (Ernawati et al., 2021). *API vision text* atau *text recognition firebase* dapat menterjemahkan teks atau angka yang ada pada gambar menjadi teks digital agar dapat diolah kembali (Chavan et al., 2020). Agar hasil OCR dapat dimunculkan keluaran suara maka diperlukan teknologi yang dapat merubah teks atau kata-kata yang dimasukkan pengguna kedalam sebuah aplikasi dengan platform tertentu menjadi bentuk keluaran suara (Fitriawati et al., 2020). *Google engine text to speech* adalah salah satu *tool* yang dapat digunakan untuk proses *text to speech* (Apriyanti & Wahyu Widodo, 2016).

Penelitian pertama oleh (Apriyanti & Wahyu Widodo, 2016) berhasil mengimplementasikan OCR pada aplikasi *text to speech* dengan bahasa Inggris sebagai keluaran suaranya. Penelitian kedua oleh (Muclis & Somantri, 2021) berhasil mengimplementasikan OCR pada aplikasi android sebagai alat bantu terjemah tulisan bahasa asing dari gambar. Penelitian ketiga oleh (Nugroho, Firmanda Mulyawan Kharisma & Wardhono, 2019) berhasil mengembangkan aplikasi pembelajaran kanji menggunakan *Firestore* ML Kit dengan *API text recognition*. Penelitian keempat oleh (Permana & Budayawan, 2020) berhasil mengembangkan aplikasi

pengklasifikasi teks ringkasan karya ilmiah menggunakan *Tensor Flow*. Penelitian kelima oleh (Zulkarnain et al., 2020) mengembangkan aplikasi pengenalan karakter dari suatu gambar dengan menggunakan *Firestore* ML Kit.

Pada penelitian ini, peneliti membangun aplikasi *text to speech* dengan menggunakan *Firestore* ML Kit sebagai pengolah OCR dan pendeteksi bahasa pada sampel gambar. Selain *text to speech*, peneliti juga menggabungkan dan melengkapi fitur aplikasi yang tidak dilakukan peneliti sebelumnya diantaranya *save file*, terjemah ke bahasa Indonesia, *text to speech* dengan 34 dialek bahasa dan salin teks ke *clipboard*. Metode yang digunakan pada penelitian ini adalah *Extreme Programming* dengan pemodelan sistem *Unified Modelling Language*.

Tujuan penelitian ini adalah aplikasi *text to speech* berbasis android yang dapat membantu pengguna sebagai alat bantu pembelajaran bahasa asing dan alat digitalisasi teks.

2. METODE PENELITIAN

Metode yang digunakan pada penelitian ini adalah *Extreme Programming* (XP). XP adalah pendekatan *agile development* yang berbasis *iterative development* (Ahmad et al., 2020; Darma et al., 2016). Metode XP menggunakan pendekatan berorientasi objek dengan sasaran tim berskala kecil sampai menengah. Metode ini cukup sederhana dan merupakan salah satu metode tangkas yang sering digunakan (Supriyatna, 2018). Tahapan metode XP diantaranya tahap *planning*, *design*, *coding*, dan *testing* (Carolina et al., 2019; Rusdiana, 2018).

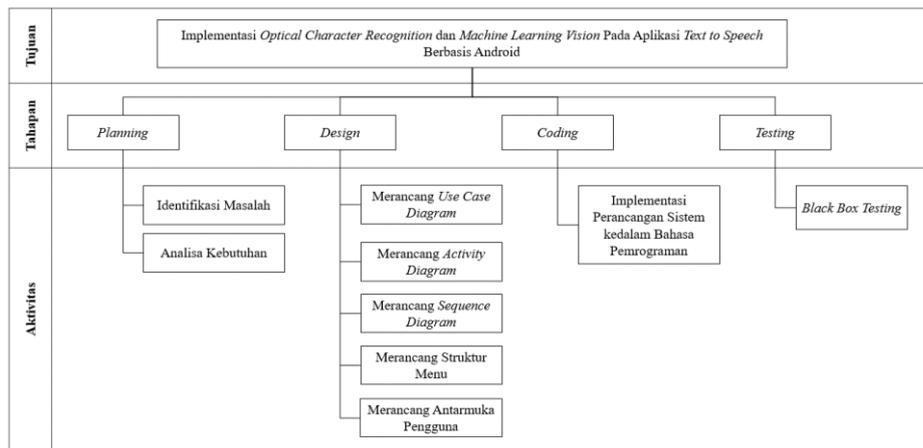
Pada tahap *planning*, dilakukan kegiatan perencanaan identifikasi masalah dan kebutuhan.

Pada tahap *design*, melakukan perancangan model sistem dan antarmuka. Dalam memodelkan sistem menggunakan UML yaitu bahasa pemodelan sistem berorientasi objek. UML dapat digunakan untuk menyederhanakan permasalahan kompleks sehingga lebih mudah dipahami dan dipelajari.

Pada tahap *coding*, melakukan penerapan model kedalam bahasa pemrograman yang digunakan.

Pada tahap *testing*, menguji aplikasi yang sudah selesai apakah sesuai dengan kebutuhan pengguna. Dalam pengujiannya menggunakan *black box testing*.

Berdasarkan metode yang digunakan, maka dibuat *work breakdown structure* yang disajikan pada Gambar 1.



Gambar 1. Work Breakdown Structure

3. HASIL DAN PEMBAHASAN

Berdasarkan metode penelitian yang digunakan maka hasil dari penelitian dapat dijelaskan sebagai berikut:

3.1.Planning

3.1.1. Identifikasi Masalah

Berdasarkan sistem yang sudah berjalan dan penelitian sebelumnya, ditemui beberapa permasalahan yang diantaranya:

1. Proses pengolahan teks dari objek tercetak dan gambar dilakukan dengan cara mengetik ulang dan akan memerlukan waktu yang lama jika teks tersebut sangat banyak.
2. Pada penelitian sebelumnya, aplikasi *text to speech* belum bisa mengeluarkan suara dengan dialek bahasa yang berbeda.

3.1.2. Analisa Kebutuhan

Kebutuhan yang akan diterapkan pada aplikasi berdasarkan permasalahan yang ditemui dapat diuraikan sebagai berikut:

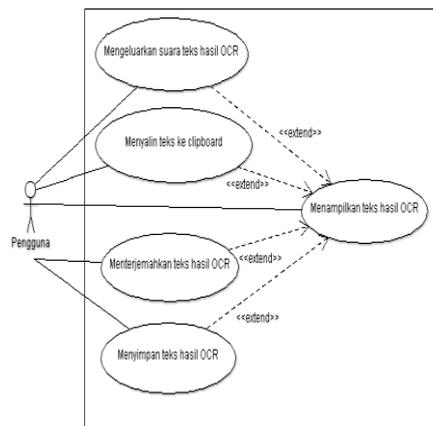
1. Aplikasi dapat menampilkan teks hasil *scan* gambar.
2. Aplikasi dapat menyalin teks hasil *scan* gambar untuk digunakan kembali di aplikasi pengolahan teks.
3. Aplikasi dapat mendeteksi bahasa yang digunakan pada teks hasil *scan* gambar kemudian dirubah menjadi keluaran suara dengan dialek bahasa hasil deteksi.
4. Aplikasi dapat menterjemahkan teks hasil *scan* gambar ke bahasa Indonesia.
5. Aplikasi dapat menyimpan teks hasil *scan* gambar menjadi file .txt.

3.2. Design

Setelah kebutuhan aplikasi didapatkan berdasarkan analisa kebutuhan, selanjutnya dibuat model sistemnya. Ditahap ini dilakukan beberapa aktivitas yang diantaranya:

3.2.1. Merancang Use Case Diagram

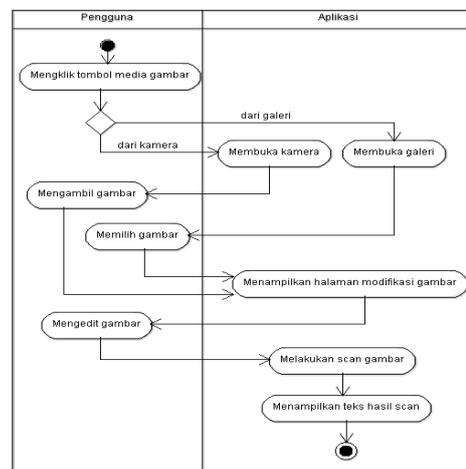
Pada Gambar 2 disajikan *use case diagram* aplikasi *text to speech* sebagai gambaran model sistemnya.



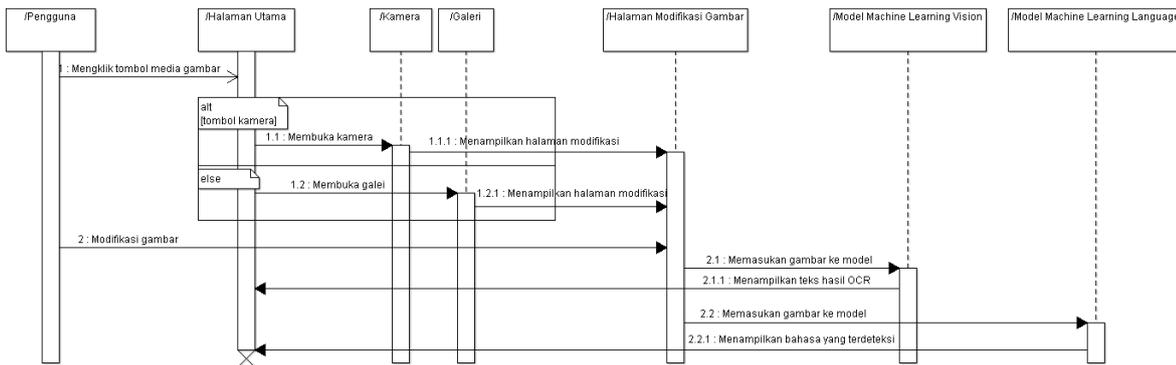
Gambar 2. Use Case Diagram Aplikasi Text to Speech

3.2.2. Merancang Activity Diagram

Pada Gambar 3 disajikan *activity diagram* untuk menampilkan teks hasil OCR.



Gambar 3. Activity Diagram Menampilkan Teks Hasil OCR



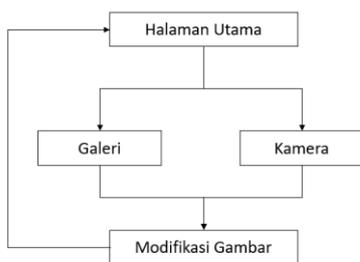
Gambar 4. Sequence Diagram Menampilkan Teks Hasil OCR

3.2.3. Merancang Sequence Diagram

Pada Gambar 4 disajikan *sequence diagram* sebagai gambaran model sistem nya.

3.2.4. Merancang Struktur Menu

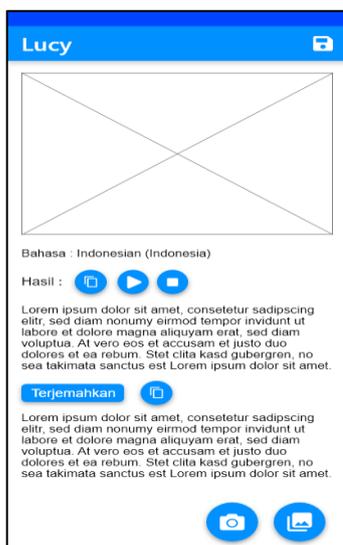
Pada Gambar 5 disajikan struktur menu aplikasi *text to speech*.



Gambar 5. Struktur Menu Aplikasi *Text to Speech*

3.2.5. Merancang Antarmuka Pengguna

Pada Gambar 6. disajikan antarmuka halaman utama aplikasi *text to speech*.



Gambar 6. Antarmuka Halaman Utama Aplikasi *Text to Speech*

3.3. Coding

Pada tahap *coding*, desain yang sudah dibuat pada tahap *design* yaitu diagram - diagram, struktur menu dan antarmuka diimplementasikan ke bahasa pemrograman *dart* menggunakan *framework flutter*. Untuk pembuatan *layout* utama mengikuti desain yang telah dibuat yaitu halaman utama, untuk *layout* galeri dan kamera menggunakan *layout provider* bawaan dari sistem operasi android, sedangkan *layout* modifikasi gambar berasal dari *library image cropper*. Untuk konfigurasi aplikasi android dibatasi untuk sistem operasi android level API 21 untuk minimalnya dan level API 30 untuk maksimalnya.

Untuk pengolahan OCR, API *Firebase ML Kit* yang digunakan adalah *vision text* sebagai *text recognition* dan *language identifier* sebagai pendeteksi bahasa yang digunakan pada hasil *scan* OCR. Google engine *text to speech* digunakan sebagai *engine* untuk merender suara dengan dialek bahasa tertentu dari teks hasil *scan* OCR.

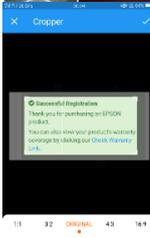
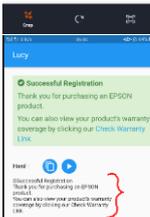
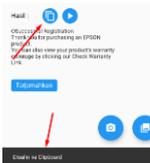
3.4. Testing

Metode *testing* yang digunakan adalah *black box testing* untuk menguji fungsionalitas aplikasi. Fokus utama dari pengujian ini adalah menguji fitur aplikasi apakah sudah berjalan dengan baik atau ditemui masalah dan apakah ada yang belum sesuai dengan kebutuhan (Cholifah et al., 2018). Untuk pengujian menggunakan *emulator* android dan *device* android. Sampel yang digunakan dalam pengujian pembacaan teks diambil dari berbagai objek yang mengandung teks. Hasil dari pengujian ini disajikan pada Tabel 1.

Berdasarkan hasil pengujian aplikasi pada Tabel 1. maka dapat disimpulkan semua fitur aplikasi sudah berjalan dengan semestinya.

Pada penelitian ini, aplikasi dapat digunakan untuk mendeteksi tulisan tangan yang pada penelitian (Apriyanti & Wahyu Widodo, 2016; Permana & Budayawan, 2020; Zulkarnain et al., 2020) tidak ada karena hanya menguji tulisan mesin saja.

Tabel 1. Pengujian Aplikasi

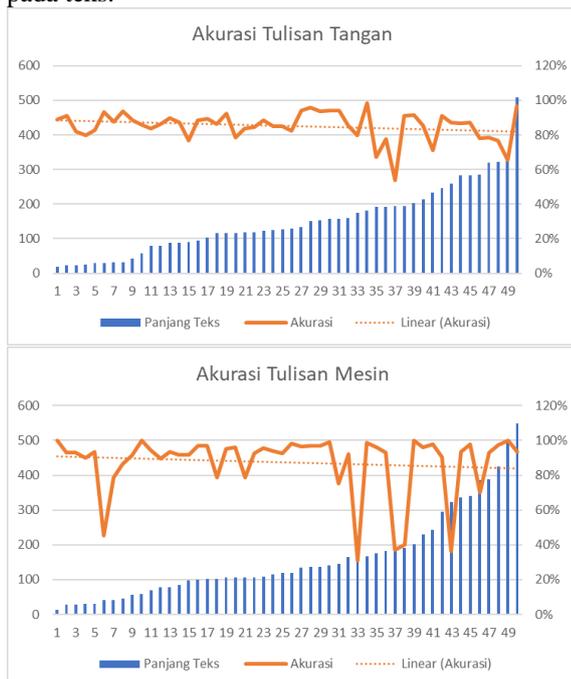
No	Fitur Uji	Skenario Uji	Hasil yang diharapkan	Hasil Uji	Keterangan
1	Ambil gambar dari kamera	Pengguna menekan tombol untuk mengambil gambar dari halaman kamera	Menampilkan kamera dan pengguna dapat melakukan pengambilan gambar		Diterima
2	Ambil gambar dari galeri	Pengguna menekan tombol untuk mengakses gambar dari halaman galeri	Menampilkan halaman galeri dan pengguna dapat memilih gambar		Diterima
3	Modifikasi gambar	Setelah memilih gambar, melakukan editing gambar untuk menyeleksi bagian yang akan discan	Gambar dapat dipotong, dirubah rotasinya dan discaling		Diterima
4	Menampilkan teks OCR	Setelah pengguna mengkonfirmasi gambar yang dimodifikasi, gambar discan dan ekstrak kata-katanya	Teks hasil deteksi dari gambar dapat diekstrak dan ditampilkan di halaman utama		Diterima
5	Text to speech	Pengguna mengklik tombol play dan aplikasi Mengeluarkan suara berdasarkan teks dan dialek bahasa yang terdeteksi	Dapat mengeluarkan dialek suara yang berbeda-beda tergantung dari bahasa yang terdeteksi		Diterima
6	Salin teks ke clipboard	Mengklik tombol copy	Menampilkan notifikasi berhasil dicopy dan hasil dari copy dapat dipaste kan di aplikasi pengolah teks		Diterima
7	Terjemah ke bahasa Indonesia	Mengklik tombol terjemahkan	Dapat menterjemahkan teks hasil deteksi ke bahasa Indonesia dan menampilkannya		Diterima
8	Simpan teks OCR	Mengklik tombol simpan	Teks hasil OCR disimpan menjadi file dengan format .txt ke folder download		Diterima

Fitur aplikasi juga kurang lengkap, diantaranya ada yang hanya menerapkan *text to speech*, *save to file .txt*, *salin teks ke clipboard*, serta *terjemah bahasa*

Inggris ke bahasa Indonesia dan sebaliknya. Sedangkan pada penelitian ini menggabungkan semua fitur yang ada pada penelitian sebelumnya.

Dialek bahasa untuk *text to speech* pada penelitian (Apriyanti & Wahyu Widodo, 2016) hanya menggunakan bahasa Inggris, sedangkan pada penelitian ini sudah mendukung 34 dialek bahasa.

Untuk mempersiapkan aplikasi agar dapat digunakan oleh masyarakat, dilakukan pengujian akurasi pengenalan teks dari tulisan tangan dan tulisan mesin. Pengujian ini menggunakan 50 sampel teks dengan panjang karakter yang berbeda-beda. Akurasi ini dihitung dari membandingkan jumlah karakter yang dikenali dengan jumlah total karakter pada teks.



Gambar 7. Grafik nilai akurasi tulisan tangan dan tulisan mesin

Berdasarkan hasil pengujian pada gambar 7 maka didapatkan rata-rata persentase akurasi untuk tulisan tangan sebesar 85,25%, sedangkan rata-rata persentase akurasi untuk tulisan mesin sebesar 87,35%. Dengan nilai akurasi yang baik ini, aplikasi siap untuk dapat dipergunakan untuk membantu proses pembelajaran bahasa asing.

Aplikasi ini selanjutnya diuji-cobakan pada grup pembelajaran bahasa asing di daerah Kabupaten Garut, dan dilakukan kuisisioner pada 32 orang peserta pelatihan yang didominasi oleh mahasiswa. Kuisisioner ini mempertanyakan nilai fitur dan nilai manfaat dari aplikasi yang telah dibangun. Dari hasil kuisisioner ini disimpulkan bahwa 71,88% memberikan nilai 4 untuk kelengkapan fitur dan 62,5% memberikan nilai 5 untuk manfaat dari aplikasi. Berdasarkan kesimpulan tersebut, aplikasi sudah cukup layak untuk digunakan oleh lebih banyak pengguna.

4. KESIMPULAN

Berdasarkan hasil dan pembahasan maka dapat disimpulkan bahwa: a) aplikasi *text to speech* sudah berjalan dengan semestinya sesuai hasil pengujian; b)

Aplikasi ini dilengkapi dengan fitur *save to file .txt*, salin teks ke *clipboard*, terjemah ke bahasa Indonesia dan dapat mengeluarkan 34 dialek bahasa yang berbeda tergantung bahasa yang terdeteksi; c) Nilai akurasi dari pengenalan teks berupa tulisan tangan rata-rata sebesar 85,25%, sedangkan rata-rata persentase akurasi untuk tulisan mesin sebesar 87,35%, serta penilaian dari penggunaan aplikasi ini 71,88% memberikan nilai 4 untuk kelengkapan fitur dan 62,5% memberikan nilai 5 untuk manfaat; dan d) dari hasil pengujian, aplikasi ini siap digunakan oleh pengguna sebagai alat bantu pembelajaran bahasa asing dan alat bantu digitalisasi teks. Sedangkan untuk pengembangan aplikasi berikutnya dapat menggunakan model *machine learning vision* untuk deteksi teks versi online atau model yang lebih bagus, dan dapat mengenali jenis teks selain teks latin saja.

DAFTAR PUSTAKA

- AHMAD, I., BORMAN, R. I., FAKHRUROZI, J., & CAKSANA, G. G. 2020. Software Development Dengan Extreme Programming (XP) Pada Aplikasi Deteksi Kemiripan Judul Skripsi Berbasis Android. *INOVTEK Polbeng - Seri Informatika*, 5(2), 297. <https://doi.org/10.35314/isi.v5i2.1654>
- APRIYANTI, K., & WAHYU WIDODO, T. 2016. Implementasi Optical Character Recognition Berbasis Backpropagation untuk Text to Speech Perangkat Android. *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, 6(1), 13. <https://doi.org/10.22146/ijeis.10767>
- CAROLINA, I., PARDEDE, A. M. H., & SUPRIYATNA, A. 2019. Penerapan Metode Extreme Programming Dalam Perancangan Aplikasi Perhitungan Kuota Sks Mengajar Dosen. *Jurnal IKRA-ITH Informatika*, 3(1), 106–113. <https://doi.org/10.31227/osf.io/se6f9>
- CHAVAN, K., BHOIR, R., GAWADE, N., & KHOT, S. 2020. Number / Text Translate from Image. *International Research Journal of Engineering and Technology (IRJET)*, 1886–1889.
- CHERIET, M., KHARMA, N., LIU, C.-L., & SUEN, C. Y. 2007. Character Recognition Systems. In *Character Recognition Systems*. John Wiley & Sons. <https://doi.org/10.1002/9780470176535>
- CHOLIFAH, W. N., YULIANINGSIH, Y., & SAGITA, S. M. 2018. Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phoneyap. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, 3(2), 206. <https://doi.org/10.30998/string.v3i2.3048>
- DARMA, D. B., JENDERAL, J., & YANI, A. 2016. Extreme Programming Sebagai Metode Pengembangan E-Keuangan Pada Pondok

- Pesantren Qodratullah. *Jurnal Ilmiah MATRIK*, 3, 163–178.
- ERNAWATI, S., WATI, R., & MAULANA, I. 2021. Penerapan Model Fountain Untuk Pengembangan Aplikasi Text Recognition dan Text To Speech berbasis Android Menggunakan Flutter. *Prosiding Snast*, 178–186.
- FITRIAWATI, L. S., BIJAKSANA, A., NEGARA, P., & NYOTO, R. D. 2020. Implementasi Text To Speech Pada Website Menggunakan Metode Shallow Parsing. *Jurnal Nasional Informatika Dan Teknologi Jaringan*, 1.
- HUSNA, E. U., ANSHORI, M., & NISA', K. 2021. Strategi Pembelajaran Program Bahasa Arab dan Bahasa Inggris Untuk Meningkatkan Daya Saing Alumni Di Era Revolusi Industri 4.0 Di Pondok. *Journal of Education and Management Studies*, 4(1), 51–58.
- MUCLIS, P. A., & SOMANTRI, S. 2021. Implementasi Text Recogniter yang Diterjemahkan ke Bahasa Lain dengan Firebase ML-Kit Berbasis Android. *Jurnal Informatika Universitas ...*, 6(2), 414–420.
- NUGROHO, FIRMANDA MULYAWAN KHARISMA, A. P., & Wardhono, W. S. 2019. Pengembangan Aplikasi Pembelajaran Kanji menggunakan MLKit Text Recognition, Text-to-Speech dan Kanji Alive API (Studi Kasus: Kun-Yomi pada JLPT N5-N3). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 3(6), 5421–5429.
- PERMANA, A., & BUDAYAWAN, K. 2020. Aplikasi Android Pengklasifikasi Semantik Teks Menggunakan Tensorflow Lite Pada Ringkasan Karya Ilmiah. *Jurnal Vocational Teknik Elektronika Dan Informatika*, 8(4).
- RUSDIANA, L. 2018. Extreme programming untuk rancang bangun aplikasi pengelolaan surat keterangan kependudukan. *Jurnal Ilmiah Teknologi Sistem Informasi*, 4(1), 49–55. <https://doi.org/10.26594/register.v4i1.1191>
- SUPRIYATNA, A. 2018. Metode Extreme Programming Pada Pembangunan Web Aplikasi Seleksi Peserta Pelatihan Kerja. *Jurnal Teknik Informatika*, 11(1), 1–18. <https://doi.org/10.15408/jti.v11i1.6628>
- ZULKARNAIN, T., SUHARYANTO, & ANTON. 2020. Perancangan Aplikasi Optical Character Recognition Berbasis Backpropagation Pada Perangkat Mobile. 14(2), 195–202.

Halaman ini sengaja dikosongkan