

## AUTOMATION TESTING TOOL DALAM PENGUJIAN APLIKASI BELAJAR TAJWID PADA PLATFORM ANDROID

Herlinda<sup>1</sup>, Dona Katarina<sup>2</sup>, Erlin Windia Ambarsari<sup>3</sup>

Informatika, Universitas Indraprasta PGRI

linda\_said72@ymail.com<sup>1</sup>, donakatarina@yahoo.com<sup>2</sup>, erlinunindra@gmail.com<sup>3</sup>

Submitted November 27, 2019; Revised November 30, 2019; Accepted Desember 1, 2019

### Abstrak

Tidak seperti pada *Black Box Testing* yang biasanya dilakukan secara manual, maka di dalam penelitian ini menggunakan *automation testing tool*, yaitu Katalon. Katalon digunakan untuk pengujian dengan menggunakan *test case*. *Test case* ini dilakukan dengan cara melakukan *record* dan *playback*, berarti perlu merekam setiap respon dalam satu event pada aplikasi yang digunakan. Tujuannya adalah untuk menganalisis efektivitas aplikasi yang sudah dibangun, yang kemudian apakah *Automation Testing Tool* Katalon berdaya guna dibandingkan pengujian manual. Oleh karena itu, akan dapat diketahui kekurangan dari penggunaan Katalon terutama pada aplikasi Android. Langkah-langkah yang dilakukan dalam pengujian aplikasi tajwid adalah Merekam *test case* dengan perangkat *mobile*, mengkonfigurasi dan merekam langkah pengujian *mobile*, menjalankan hasil *test case*, dan melihat *log* eksekusi. Dari penelitian didapatkan terjadi tiga kesalahan didalam aplikasi, diantaranya ketidaksesuaian ukuran layar dalam mode *landscape*, terdapatnya kegagalan eksekusi dikarenakan Katalon tidak dapat mendeteksi objek yang tidak masuk dalam tangkap layar *Device View*, dan objek suara yang tidak berhenti jika sudah pindah halaman layar.

**Kata Kunci:** *Black Box Testing, Katalon, Android, Test Case*

### Abstract

*Dissimilar in Black Box Testing, which is usually manually, this research uses an automation testing tool, namely Katalon. The Katalon used for testing using a test case. A test case by recording and playback; it needs to record every response in one event in the application used. The aim is to analyze the effectiveness software that has built and whether the Katalon Automation Testing Tool is efficient if compared to manual testing. Therefore, it is known the shortcomings of the use of Katalon, especially on Android applications. The steps taken in testing the Tajweed application are recording the test case with a mobile device, configuring and documenting the actions of the portable test, running the results of the test case, and viewing the execution log. The research found three errors that occurred in the application. There is the screen size mismatch in landscape mode, a failure of execution because the Katalon could not detect objects that were not captured by the Device View and sound objects that did not stop if the screen page had switched over.*

**Key Words:** *Black Box Testing, Katalon, Android, Test Case*

## 1. PENDAHULUAN

Pada era industri 4.0, aplikasi *mobile* sudah banyak mendominasi *business intelligent* yang dilakukan oleh para *startup* sebagai pengembang dengan memanfaatkan *platform* Android dan IOS. Contohnya seperti bukalapak, tokopedia, gojek, dan traveloka yang berbasis *mobile* dengan

mengembangkan *Artificial Intelligent* untuk memahami keinginan *user* sebagai pelanggannya. Ditambah pula dengan berjamurnya Fintech P2P *Lending* yang memanfaatkan teknologi tersebut, menelusuri bagaimana perilaku calon kreditur untuk menentukan penerimaan pengajuan kredit.

Belum lagi popularnya *e-sport* seperti PUBG, Mobile Legends, Dota 2, Pro Evaluation Soccer (PES), Arena of Valor, dan Clash Royale yang dipertandingkan di SEA Games, ASIAN Games, dan turnamen lainnya. Dengan kata lain, *end user* diberi kemudahan dengan dimanjakan oleh ragamnya aplikasi. Namun, perlu diperhatikan pula untuk mengembangkan aplikasi berbasis Android, sudah pasti masalah *bug* tidak dapat dihindarkan. Kesalahan terjadi pada perangkat lunak yang disebabkan karena tidak sesuai dengan kebutuhan perangkat lunak [1], sehingga perlu dilakukan perbaikan.

Ketika perbaikan *bug*, dilakukan pengujian seperti memeriksa setiap fitur yang digunakan oleh aplikasi. Metode-metode yang digunakan dalam pengujian dapat dilakukan dengan *Black Box Testing*. Berdasarkan pada penelitian dari [2] bahwa *Black Box Testing* tidak memerlukan pengujian desain maupun kode program, melainkan hanya fokus terhadap fungsinya saja sesuai dengan spesifikasi yang dibutuhkan. Pengujian *Black Box Testing*, dapat dilakukan dengan manual dan otomatis. Menurut [3], pengujian secara manual tidak efektif. Perlu juga dilakukan dengan pengujian otomatis.

Berdasarkan pernyataan dari [4], yang dimana secara umumnya pengujian manual merupakan aktivitas sistematis dan mengikuti proses. Proses tersebut melibatkan para pakar untuk menganalisis kebutuhan, dengan mempersiapkan rencana pengujian, mengatur lingkungan pengujian, mengidentifikasi skenario pengujian, mempersiapkan pengujian kasus dan data, pengujian eksekusi, dan laporan cacat. Namun, pengujian ini tidak efisien jika dilakukan pengujian regresi (berulang-ulang). Misalkan pada penelitian dari [5] memerlukan 16 orang anggota BRIMOB untuk melakukan validasi dalam pengujian aplikasi *Action & Strategy*. Sedangkan pada pengujian otomatis tidak

melibatkan pakar untuk validasi dan dapat mengendalikan pengujian eksekusi dengan kemampuan menjalankan banyak pengujian dengan waktu singkat, kompatibilitas yang berarti dapat menjalankan di lingkungan berbeda. Selain itu, pengujian otomatis dapat mempercepat pengujian unit, pengujian API, dan pengujian GUI.

Oleh karena itu, pada penelitian ini dilakukan menggunakan pengujian otomatis. Alat yang digunakan dalam pengujian otomatis atau disebut dengan *Automation Testing Tool*, yaitu Katalon. Dalam penelitian [6], Katalon digunakan untuk pengujian *e-commerce*. Sampel yang diambil adalah Tokopedia, JD. ID, dan Bukalapak, dengan menggunakan *test case*. *Test case* ini dilakukan dengan cara melakukan *record* dan *playback*, berarti perlu merekam setiap respon dalam satu *event* pada *e-commerce* yang dilakukan pengguna.

Pengujian dengan menggunakan Katalon pada penelitian ini, dilakukan kepada aplikasi dengan *platform* Android. Sampel yang digunakan adalah aplikasi Tajwid yang dibangun oleh [7] dengan melakukan *test case*. Tujuannya adalah untuk menganalisis efektivitas aplikasi yang sudah dibangun, yang kemudian apakah *Automation Testing Tool* Katalon berdaya guna dibandingkan pengujian manual. Oleh karena itu, akan dapat diketahui kekurangan dari penggunaan Katalon terutama pada aplikasi Android.

## 2. METODE PENELITIAN

Katalon yang dikembangkan oleh Katalon LLC bertujuan sebagai pemberi solusi otomatis yang terjangkau, karena berbasis *open source* dan dibuat sesimpel mungkin, sehingga dengan dibangunnya Katalon Studio dapat membantu QA (*Software Quality Assurance*) baik untuk QA pemula maupun profesional untuk melakukan pengujian dengan mudah. Pengujian pada

Katalon mencakup *Web UI, API, Mobile* dan *Scripting*. Berbeda yang dilakukan oleh [6], dimana dapat langsung menggunakan Katalon dalam pengujiannya karena pengujian dilakukan adalah GUI halaman *web*. Pada penelitian ini, pengujian yang dilakukan adalah pengujian aplikasi berbasis Android (*Mobile*), sehingga perlu dilakukan *setup* terlebih dahulu. Sesuai pada pedoman dari [8], alat-alat yang digunakan untuk *setup* adalah sebagai berikut:

1) Node.js

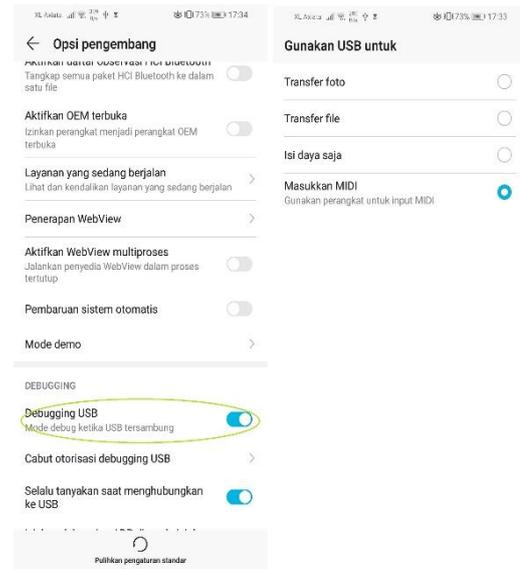
Node.js dibangun sebagai aplikasi berbasis *web* dan ditulis dalam sintaks Javascript. Node.js digunakan untuk melengkapi peran JavaScript (*an asynchronous event-driven JavaScript runtime*) sehingga bisa berjalan di sisi *server*. Bahkan untuk dijalankan tanpa menggunakan program *server web*, dikarenakan mempunyai pustaka *server HTTP* sendiri.

2) Appium

Appium digunakan sebagai alat otomatisasi untuk aplikasi *native* (asli), *mobile web*, ataupun *hybrid* untuk *platform* iOS serta android. Appium dapat dijalankan pada simulator ataupun perangkat asli (*smartphone*) yang berbasis *open source*.

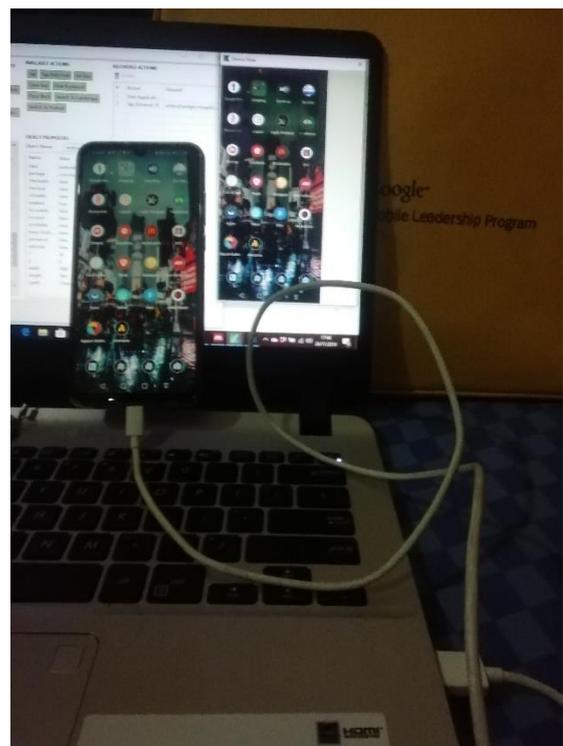
3) Perangkat Asli

Katalon memerlukan *Operating System version* yang terdapat dalam iOS atau Android. Oleh karena itu, *Smartphone* yang digunakan sudah terdapat instalasi *USB driver* dan terdeteksi melalui opsi pengembang (Gambar 1) dengan mengkonfigurasi *Debugging USB* didalam *smartphone* (fitur opsi pengembang berbeda-beda, tergantung dari *smartphone* yang digunakan). Hasil konfigurasi dari opsi pengembang tersebut, membuat *smartphone* terkoneksi pada Katalon yang dapat dilihat pada Gambar 2.



Sumber : Dokumen Pribadi

**Gambar 1. Konfigurasi Pada Opsi Pengembang**



Sumber : Dokumen Pribadi

**Gambar 2. Koneksi Smartphone dengan Katalon dari Konfigurasi Gambar 1.**

Setelah dilakukan *setup* alat-alat untuk pengujian Android, maka yang dilakukan adalah melakukan pengujian *test case* pada aplikasi Tajwid dengan mengikuti langkah-langkah sebagai berikut:

- 1) Merekam *Mobile Utility*[9]  
Merekam atau *record* tidak dapat dilakukan jika belum melakukan konfigurasi terlebih dahulu, poin penting yang dilakukan dalam konfigurasi antara lain:

**Tabel 1. Konfigurasi Rekaman *Mobile***

Bidang	Penjelasan
<p><i>Device Name</i></p> <p>Device Name: HUAWEI HRY-AL00a (Android 9)</p>	<p><i>Smartphone</i> yang terhubung melalui USB <i>driver</i> akan ditangkap objeknya dan ditampilkan dalam daftar alat, misalkan pada penelitian ini yang terdeteksi adalah HUAWEI HRY-AL00a (android 9). Dengan kata lain <i>driver</i> yang digunakan adalah <i>smarphone</i> HUAWEI seri HRY-AL00a dengan versi android 9 (Pie).</p>
<p><i>Application Type</i></p> <p>Start with: Application File</p>	<p>Didukung oleh perangkat <i>mobile</i> asli (lokal) dan perangkat <i>cloud-based</i> Kobiton. Kobiton berfungsi untuk mempercepat pengiriman dan pengujian aplikasi berbasis <i>mobile</i> dalam perangkat asli</p>
<p><i>Application File</i></p> <p>Application File: C:\Users\erlin\OneDrive\Documents\apl Android BelajarTajwid.apk</p>	<p>Menunjukkan alamat dari sebuah File. File yang digunakan pada pengujian ini adalah BelajarTajwid.apk dengan alamat C:\Users\erlin\OneDrive\Documents\apl Android BelajarTajwid Herlinda\BelajarTajwid.apk</p>

Sumber : Katalon *Team's Document* [9]

Kemudian memulai perekaman dengan menekan tombol *Start*, Katalon akan mengaktifkan Appium dan ditransfer ke dalam *smartphone* sehingga Android dapat berinteraksi dengan beberapa sistem API. Setelah itu, Katalon menangkap semua objek yang ada dalam aplikasi Tajwid dan tersimpan pada *All Objects* dalam bentuk hirarki. Jika memilih elemen objek apapun pada hirarki dalam *All Objets*, maka akan tersorot didalam

*Device View*. *Device View* berfungsi sebagai penangkap layar aplikasi dari *smartphone* yang digunakan.

Setelah memilih elemen objek, dapat dilakukan eksekusi setiap fitur yang digunakan dalam aplikasi dengan memilih setiap *action* pada *Available Actions*.

- 2) Mengeksekusi *test case* yang sudah digeneralisasi di salah satu perangkat yang didukung yaitu HUAWEI HRY-AL00a (android 9).

### 3. HASIL DAN PEMBAHASAN

Berdasarkan penjelasan sebelumnya mengenai *Available Actions*, yang mempunyai fungsi sebagai eksekusi setiap *action* dimana merupakan tombol-tombol fitur aplikasi tajwid yang nantinya akan disimpan kedalam *Recorded Actions* seperti pada gambar dibawah ini.

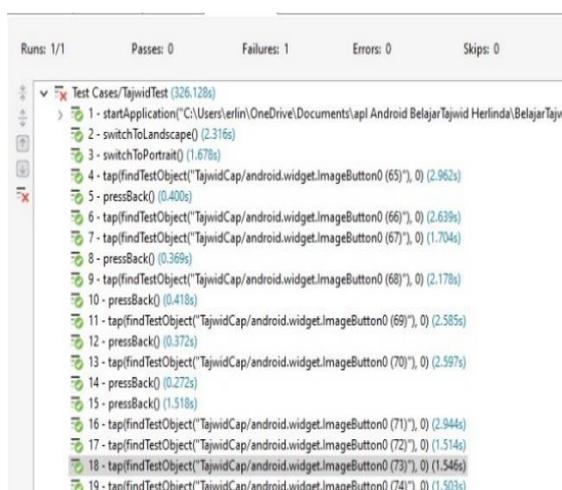
RECORDED ACTIONS		
Delete		
#	Action	Element
1	Start Applicati...	
2	Switch To Lan...	
3	Switch To Port...	
4	Tap [timeout: 0]	android.widget.ImageBu...
5	Press Back	
6	Tap [timeout: 0]	android.widget.ImageBu...
7	Tap [timeout: 0]	android.widget.ImageBu...
8	Press Back	
9	Tap [timeout: 0]	android.widget.ImageBu...
10	Press Back	
11	Tap [timeout: 0]	android.widget.ImageBu...
12	Press Back	
13	Tap [timeout: 0]	android.widget.ImageBu...
14	Press Back	
15	Press Back	
16	Tap [timeout: 0]	android.widget.ImageBu...

**Gambar 3. Recorded Actions Aplikasi Tajwid**

*Recorded Action* sebagai dasar analisa pengujian dalam bentuk *script testing* yang diambil dari hasil rekaman *mobile utility* dan eksekusi *test case*.

```
Mobile.startApplication('C:\\Users\\erlin\\OneDrive\\Documents\\apl Android BelajarTajwid Herlinda\\BelajarTajwid.apk',  
true)  
Mobile.switchToLandscape()  
Mobile.switchToPortrait()  
Mobile.tap(findTestObject('TajwidCap/android.widget.  
ImageButton0 (65)'), 0)  
Mobile.pressBack()  
Mobile.tap(findTestObject('TajwidCap/android.widget.  
ImageButton0 (66)'), 0)  
Mobile.tap(findTestObject('TajwidCap/android.widget.  
ImageButton0 (67)'), 0)  
Mobile.pressBack()  
Mobile.tap(findTestObject('TajwidCap/android.widget.  
ImageButton0 (68)'), 0)  
Mobile.pressBack()  
Mobile.tap(findTestObject('TajwidCap/android.widget.  
ImageButton0 (69)'), 0)  
Mobile.pressBack()
```

Berdasarkan dari eksekusi tersebut, dilakukan *Job Progress* untuk menganalisa *response time* pada setiap objek didalam aplikasi tajwid. *Response time* tersebut tergantung dari respon *smartphone* saat membuka setiap objek sehingga jika *Job Progress* dilakukan berulang, maka hasil *response time* pun akan berbeda dari *response time* sebelumnya. Responsif dari aplikasi tajwid dapat dilihat pada *Log Viewer* seperti gambar berikut:



Gambar 4. Log Viewer aplikasi tajwid

Pada aplikasi tajwid membutuhkan 326,128 detik atau rata-rata sekitar 26,128 detik sampai dengan 5 menit untuk setiap objek didalam aplikasi. Pengujiannya pun membutuhkan sekitar 121 langkah dari mulai membuka aplikasi sampai ditutup kembali.

Diantara 121 langkah tersebut terdapat objek yang paling penting untuk dianalisa pengujiannya, yaitu pada fitur-fitur di dalam aplikasi tajwid dimana lebih banyak menggunakan *template* dan tidak membutuhkan *database*. Objek-objek yang dianalisa diantaranya adalah:

- 1) Fitur *About*, yang merupakan informasi dari aplikasi tajwid.
- 2) Fitur *Mahraj*, merupakan ilustrasi tempat keluarnya huruf. Mahraj terdiri dari objek Mahraj Rongga Mulut, Mahraj Tenggorokan, Mahraj Lidah, dan Mahraj Bibir
- 3) Fitur *Tajwid*, merupakan bagaimana cara mengucapkan huruf-huruf yang terdapat dalam Al-Quran maupun bukan. Tajwid sendiri terdiri dari objek Tanwin (Izhar. Idgham, Ikhfa, Iqlab), Mim Mati (Ikhfa Syafawi, Idgham Mislain, Izhar Syafawi), Hukum Mad (Mad Thobi'i, Mad Wajib Muttasii, Mad Jaiz Munfasil, Mad Arid Lisukun), Hukum Idgham (Idgham Mutamatsilain, Idgham Mutajanisain, Idgham Mutaqaribain), dan Qalqalah (Qalqalah Kubro, Qalqalah Sugro). Ilustrasi Tajwid lebih sulit jika hanya berupa tulisan. Oleh karena itu, untuk mempermudah mendemonstrasikan objek-objek di dalam tajwid, maka dari masing-masing objek tersebut disematkan objek tombol suara.
- 4) Fitur *Kuis*; yang terdiri 18 pertanyaan, masing-masing pertanyaan terdiri dari objek pilih (*radio button*) dan objek tombol *btnsubmit* untuk eksekusi jawaban dari pertanyaan yang diajukan.



Sumber : Aplikasi Belajar Tajwid[7]

**Gambar 5. Fitur Yang Dijadikan Objek Pengujian**

Berikut adalah beberapa *response time* dari objek-objek aplikasi tajwid yang diambil dalam *test case*.

**Tabel 2. Response Time Aplikasi Tajwid**

No	Objek Utama	Respon se Time (detik)	Keterangan
1	Menjalankan Aplikasi	32,068	Aplikasi BelajarTajwid.apk Sukses, tetapi tampilan melebihi layar
2	Rubah tampilan layar ke <i>Landscape</i>	2,316	layar
3	<i>About</i>	2,962	Sukses
4	Mahraj	2,639	Sukses
5	Mahraj/ronggam <ulut< td=""> <td>1,704</td> <td>Sukses</td> </ulut<>	1,704	Sukses
6	Mahraj/tenggorokan	2,178	Sukses
7	Mahraj/lidah	2,585	Sukses
8	mahraj/bibir	2,597	Sukses
9	Tajwid	2,944	Sukses
10	tajwid/tanwin	1,514	Sukses

11	tajwid/tanwin/iz har	1,546	Sukses
12	tajwid/tanwin/iz har-suara	1,503	Sukses
13	tajwid/tanwin/id ghm	3,067	Sukses
14	tajwid/tanwin/id ghm-suara1	31,971	Gagal, Tidak ditemukan
15	tajwid/tanwin/id ghm-suara2	31,286	Gagal, Tidak ditemukan
16	tajwid/tanwin/ik hfa	2,5	Sukses
17	tajwid/tanwin/ik hfa-suara	33,027	Sukses
18	tajwid/tanwin/iq lab	2,605	Sukses
19	tajwid/tanwin/iq lab-suara	2,991	Sukses
20	tajwid/mimmati	3,136	Sukses
21	tajwid/mimmati/syafawi	1,442	Sukses
22	tajwid/mimmati/syafawi-suara	3,5	Sukses
23	tajwid/mimmati/mislain	2,294	Sukses
24	tajwid/mimmati/mislain-suara	1,478	Sukses

Berdasarkan Tabel 2, didapatkan beberapa kesalahan dari aplikasi tajwid yang diantaranya adalah tampilan layar tidak dapat menyesuaikan ketika dirubah ke dalam mode *Landscape*. Sehingga mengakibatkan tampilan aplikasi keluar dari tampilan layar *smartphone*. Oleh karena itu, perlu dilakukan pengaturan *mode screen* agar tetap dalam layar *portrait*.



**Gambar 6. Tampilan Layar Dalam Bentuk Mode Landscape**

Kemudian terjadi kegagalan pada saat menjalankan sistem, yaitu pada no. 14 tajwid/tanwin/idghm-suara1 dan 15 tajwid/tanwin/idghm-suara2. Hal ini terjadi dikarenakan *response time* lebih lama yaitu 31,971 dan 31,286 detik. Oleh karena itu, Katalon tidak dapat mendeteksi objek. Alasannya adalah pada Katalon tidak menyediakan *slide* sehingga layar tidak dapat digerakkan, dengan kata lain tangkap layar pada *Device View* bersifat statis. Karena itu, cara lain untuk dapat menjalankan sistem adalah dengan menggerakkan layar sentuh pada *smartphone* yang terkoneksi pada Katalon.

Selain kegagalan yang dijelaskan sebelumnya, ada hal yang ternyata perlu diperhatikan bahwa walaupun objek suara dikatakan sukses, pada saat berpindah halaman layar suara tetap berjalan sampai selesai. Bilamana pengguna mengaktifkan tombol suara yang lain, alhasil suara akan tumpang tindih. Contoh pada no. 17 tajwid/tanwin/ikhfa-suara dengan *response time* 33,027 bahwa respon objek suara tersebut sangat lama karena mempunyai ukuran memori yang cukup besar. Sehingga perlu dilakukan *program control* untuk mengendalikan objek suara agar saat berpindah halaman layar, suara tersebut dapat berhenti. Sehingga dalam hal ini, perlu juga dilakukan *White Box Testing* untuk menganalisa instruksi-instruksi yang dilakukan didalam pemrograman.

Berdasarkan analisa-analisa diatas, Katalon masih ada kekurangannya yaitu untuk dapat melakukan pengujian, perangkat asli tetap harus diaktifkan dan digerakkan karena pengujiannya langsung melalui koneksi dari *smartphone* dengan menggunakan kabel USB. Walaupun saat merekam menggunakan tangkap layar dari *Device View*. Namun, secara garis besar fungsinya sebagai *Automation Testing Tool* sudah dapat terpenuhi.

#### 4. SIMPULAN

Objek yang dilakukan pengujian diantaranya adalah Fitur *About*; Fitur Mahraj (Mahraj Rongga Mulut, Mahraj Tenggorokan, Mahraj Lidah, dan Mahraj Bibir); Fitur Tajwid yaitu Tanwin (Izhar. Idgham, Ikhfa, Iqlab), Mim Mati (Ikhfa Syafawi, Idgham Mislain, Izhar Syafawi); Hukum Mad (Mad Thobi'i, Mad Wajib Muttasii, Mad Jaiz Munfasil, Mad Arid Lisukun); Hukum Idgham (Idgham Mutamatsilain, Idgham Mutajanisain, Idgham Mutaqaribain); Qalqalah (Qalqalah Kubro, Qalqalah Sugro); serta yang terakhir Fitur Kuis. Selain itu, ada pula objek suara yang perlu diujikan. Aplikasi tajwid ketika diuji membutuhkan 326,128 detik atau rata-rata sekitar 26,128 detik sampai dengan 5 menit dengan 121 langkah pengujian. Dari 121 langkah tersebut, terdapat 3 kegagalan, antara lain tampilan layar tidak dapat menyesuaikan ketika dirubah ke dalam mode *Landscape*. Sehingga mengakibatkan tampilan aplikasi keluar dari tampilan layar *smartphone*, Katalon tidak dapat mendeteksi objek yaitu tajwid/tanwin/idghm-suara1 (31,971) dan tajwid/tanwin/idghm-suara2 (31,286). Alasannya adalah pada Katalon tidak menyediakan *slide* sehingga layar tidak dapat digerakkan atau dengan kata lain tangkap layar pada *Device View* bersifat statis. Terdapat objek suara tetap berjalan sampai selesai walaupun sudah berpindah halaman layar. Bilamana mengaktifkan tombol suara yang lain, alhasil suara akan tumpang tindih. Penyebabnya dikarenakan mempunyai ukuran memori yang cukup besar sehingga *response time* menjadi lebih lama. Hal ini perlu dilakukan pengujian *White Box* untuk menganalisa instruksi pemrograman di dalam aplikasi. Berdasarkan analisa, dapat disimpulkan bahwa Katalon masih ada kekurangannya yaitu untuk dapat melakukan pengujian, perangkat asli tetap harus diaktifkan dan digerakkan karena pengujiannya tergantung dari koneksi *smartphone*.

Walaupun saat merekam menggunakan *Device View*. Namun, fungsinya sebagai *Automation Testing Tool* sudah dapat terpenuhi.

#### DAFTAR PUSTAKA

- [1] N. F. Azhar and S. Rochimah, "Memprediksi Waktu Memperbaiki Bug dari Laporan Bug Menggunakan Klasifikasi Random Forest," *J. Sist. Dan Inform.*, vol. 11, no. 1, pp. 156–164, 2016.
- [2] N. Kustian, "PENGUNAAN MODEL WATERFALL DALAM PEMBUATAN APLIKASI PEMESANAN TIKET BUS," *STRING (Satuan Tulisan Ris. dan Inov. Teknol.*, vol. 4, no. 1, pp. 94–104, 2019.
- [3] M. K. MZ, "Pengujian perangkat Lunak metode Black box berbasis partitions pada aplikasi sistem informasi di sekolah," *J. Mikrotik*, vol. 6, no. 3, pp. 2–16, 2016.
- [4] K. Saravanan and E. P. C. Prasad, "Open Source Software Test Automation Tools: A Competitive Necessity," *Sch. Int. J. Manag. Dev. ISSN 2394-3378*, vol. 3, no. 6, pp. 103–110, 2016.
- [5] W. N. Cholifah, Y. Yulianingsih, and S. M. Sagita, "Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap," *STRING (Satuan Tulisan Ris. dan Inov. Teknol.*, vol. 3, no. 2, pp. 206–210, 2018.
- [6] M. M. Muhtadi, M. D. Friyadi, and A. Rahmani, "Analisis GUI Testing pada Aplikasi E-Commerce menggunakan Katalon," *Pros. Ind. Res. Work. Natl. Semin.*, vol. 10, no. 1, pp. 1387–1393, 2019.
- [7] D. Kusumawati, "Perancangan Aplikasi Pengenalan Cara Membaca Tajwid Menggunakan Android Studio Dengan Metode SDLC," Universitas Indraprasta PGRI, 2018.
- [8] Katalon Team, "Android setup on Windows | Katalon Docs." [Online]. Available: [https://docs.katalon.com/katalon-studio/videos/android\\_setup\\_windows.html](https://docs.katalon.com/katalon-studio/videos/android_setup_windows.html). [Accessed: 23-Nov-2019].
- [9] Katalon Team, "Record Mobile Utility | Katalon Docs." [Online]. Available: <https://docs.katalon.com/katalon-studio/docs/record-mobile-utility.html>. [Accessed: 24-Nov-2019].