



**IMPLEMENTASI AUTOMATION TEST DENGAN METODE BLACK  
BOX TESTING PADA APLIKASI ANDROID MENGGUNAKAN APPIUM  
DAN GHERKIN SYNTAX DI PT. XYZ**

*TUGAS AKHIR*



Louis Onike Munte  
41519110063

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS MERCU BUANA  
JAKARTA  
2021**



**IMPLEMENTASI AUTOMATION TEST DENGAN METODE BLACK  
BOX TESTING PADA APLIKASI ANDROID MENGGUNAKAN APPIUM  
DAN GHERKIN SYNTAX DI PT. XYZ**

*Tugas Akhir*

Diajukan Untuk Melengkapi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer

Oleh:

Louis Onike Munte

41519110063

UNIVERSITAS  
MERCU BUANA  
PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS MERCU BUANA  
JAKARTA  
2021

### LEMBAR PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini:

NIM : 41519110063

Nama : Louis Onike Munte

Judul Tugas Akhir : Implementasi Automation Test dengan Metode Black Box Testing pada Aplikasi Android menggunakan Appium dan Gherkin Syntax di PT. XYZ

Menyatakan bahwa Laporan Tugas Akhir saya adalah hasil karya sendiri dan bukan plagiat. Apabila ternyata ditemukan didalam laporan Tugas Akhir saya terdapat unsur plagiat, maka saya siap untuk mendapatkan sanksi akademik yang terkait dengan hal tersebut.

Jakarta, 3 Agustus 2021



Louis Onike Munte



UNIVERSITAS  
MERCU BUANA

## SURAT PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR

Sebagai mahasiswa Universitas Mercu Buana, saya yang bertanda tangan di bawah ini :

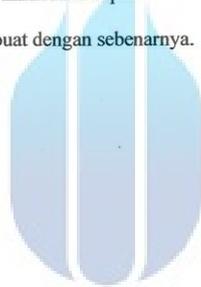
Nama Mahasiswa : Louis Onike Munte  
NIM : 41519110063  
Judul Tugas Akhir : Implementasi Automation Test dengan Metode  
Black Box Testing pada Aplikasi Android  
menggunakan Appium dan Gherkin Syntax di PT.  
XYZ

Dengan ini memberikan izin dan menyetujui untuk memberikan kepada Universitas Mercu Buana **Hak Bebas Royalti Noneksklusif** (*None-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul diatas beserta perangkat yang ada (jika diperlukan).

Dengan Hak Bebas Royalti/Noneksklusif ini Universitas Mercu Buana berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir saya.

Selain itu, demi pengembangan ilmu pengetahuan di lingkungan Universitas Mercu Buana, saya memberikan izin kepada Peneliti di Lab Riset Fakultas Ilmu Komputer, Universitas Mercu Buana untuk menggunakan dan mengembangkan hasil riset yang ada dalam tugas akhir untuk kepentingan riset dan publikasi selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.



Jakarta, 3 Agustus 2021



Louis Onike Munte

UNIVERSITAS  
MERCU BUANA

### SURAT PERNYATAAN LUARAN TUGAS AKHIR

Sebagai mahasiswa Universitas Mercu Buana, saya yang bertanda tangan di bawah ini :

Nama Mahasiswa : Louis Onike Munte  
NIM : 41519110063  
Judul Tugas Akhir : Implementasi Automation Test dengan Metode  
Black Box Testing pada Aplikasi Android  
menggunakan Appium dan Gherkin Syntax di PT.  
XYZ

Menyatakan bahwa :

1. Luaran Tugas Akhir saya adalah sebagai berikut :

No	Luaran	Jenis	Status
1	Publikasi Ilmiah	Jurnal Nasional Tidak Terakreditasi	Diajukan
		Jurnal Nasional Terakreditasi	
		Jurnal International Tidak Bereputasi	Diterima
		Jurnal International Bereputasi	
Disubmit/dipublikasikan di :	Nama Jurnal	:	
	ISSN	:	
	Link Jurnal	:	
	Link File Jurnal Jika Sudah di Publish	:	
		:	

2. Bersedia untuk menyelesaikan seluruh proses publikasi artikel mulai dari submit, revisi artikel sampai dengan dinyatakan dapat diterbitkan pada jurnal yang dituju.
3. Diminta untuk melampirkan scan KTP dan Surat Pernyataan (Lihat Lampiran Dokumen HKI), untuk kepentingan pendaftaran HKI apabila diperlukan

Demikian pernyataan ini saya buat dengan sebenarnya.

Jakarta, 3 Agustus 2021



Louis Onike Munte

UNIVERSITAS  
MERCU BUANA

## LEMBAR PERSETUJUAN PENGUJI

NIM : 41519110063  
Nama : Louis Onike Munte  
Judul Tugas Akhir : Implementasi Automation Test dengan Metode Black Box Testing pada Aplikasi Android menggunakan Appium dan Gherkin Syntax di PT. XYZ

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 3 Agustus 2021



(Achmad Kodar, Drs. MT)

UNIVERSITAS  
MERCU BUANA

## LEMBAR PERSETUJUAN PENGUJI

NIM : 41519110063  
Nama : Louis Onike Munte  
Judul Tugas Akhir : Implementasi Automation Test dengan Metode Black Box Testing pada Aplikasi Android menggunakan Appium dan Gherkin Syntax di PT. XYZ

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 3 Agustus 2021



(Sabar Rudiarto, M.Kom)

UNIVERSITAS  
MERCU BUANA

## LEMBAR PERSETUJUAN PENGUJI

NIM : 41519110063  
Nama : Louis Onike Munte  
Judul Tugas Akhir : Implementasi Automation Test dengan Metode Black Box Testing pada Aplikasi Android menggunakan Appium dan Gherkin Syntax di PT. XYZ

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 3 Agustus 2021



(Hery Derajad Wijaya, S.Kom., MM)

UNIVERSITAS  
MERCU BUANA

## LEMBAR PENGESAHAN

NIM : 41519110063  
Nama : Louis Onike Munte  
Judul Tugas Akhir : Implementasi Automation Test dengan Metode Black Box Testing pada Aplikasi Android menggunakan Appium dan Gherkin Syntax di PT. XYZ

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 3 Agustus 2021

Menyetujui,



(Harni Kusniyati, M.Kom)  
Dosen Pembimbing

Mengetahui,

UNIVERSITAS  
MERCUBUANA

(Wawan Gunawan, S.Kom., MT)  
Koord. Tugas Akhir Teknik Informatika

(Hery Derajad Wijaya, S.Kom., MM)  
Ka. Prodi Teknik Informatika

## ABSTRAK

Nama : Louis Onike Munte  
NIM : 41519110063  
Pembimbing TA : Harni Kusniyati, M.Kom  
Judul : Implementasi Automation Test dengan Metode Black Box Testing pada Aplikasi Android menggunakan Appium dan Gherkin Syntax di PT. XYZ

*Testing* adalah proses mengevaluasi sistem atau komponen dengan maksud untuk menemukan bahwa apakah sistem tersebut memenuhi persyaratan yang ditentukan atau tidak. Saat ini, di PT. XYZ, pengujian aplikasi masih dilakukan secara manual. Hal ini membuat pengujian cukup kewalahan dikarenakan dengan tersedianya sistem dalam beberapa layanan, yaitu *web*, *mobile web*, dan *mobile app*. Waktu yang dibutuhkan dalam proses pengujian lebih lama, sehingga proses rilis aplikasi juga menjadi lebih lama. Untuk itu perlu adanya peningkatan dalam proses pengujian aplikasi dengan mengotomatisasi pengujian. Tulisan ini membahas tentang implementasi pengujian otomatis pada aplikasi Android menggunakan Appium dengan format skenario uji menggunakan Gherkin *syntax*. Eksperimen dilakukan dengan menjalankan skenario uji sesuai dengan 8 kriteria pengujian yang ditetapkan yaitu waktu pengujian, bahasa pemrograman yang digunakan, dokumentasi hasil eksekusi kasus uji, dukungan proses *delay*, dukungan proses *suspend/resume*, dukungan proses *relaunch*, dukungan proses *scrolling*, serta penyimpanan perubahan selama pengujian. Hasil yang diperoleh dari eksperimen yang dilakukan, Appium memenuhi 8 kriteria pengujian tersebut. Penulis merekomendasikan penggunaan Appium dalam melakukan pengujian otomatis untuk aplikasi Android.

Kata kunci:

Pengujian Otomatis, Appium, Android, Gherkin, Skenario Uji

## ABSTRACT

Name : Louis Onike Munte  
Student Number : 41519110063  
Counsellor : Harni Kusniyati, M.Kom  
Title : Automation Test Implementation with Black Box Testing Method in Android Application Using Appium and Gherkin Syntax at PT. XYZ

Testing is an evaluation process for system or component with the intention of finding that whether the system meets the specified requirements or not. Currently, at PT. XYZ, application testing is still done manually. This made the testers quite overwhelmed because of the availability of the system in several services, namely web, mobile web and mobile applications. The time needed in the testing process is longer, so the application release process is also longer. For this reason, it is necessary to increase in the application testing process by automating testing. This paper discusses the implementation of automated testing on Android applications using Appium with a test scenario format using Gherkin syntax. Experiments were carried out by running scenario tests according to the 8 specified testing criteria, namely the time of the examiner, the programming language used, documentation of the results of the test case execution, support for the delay process, support for the suspension / resume process, support for the relaunch process, support for the scrolling process, and storage of changes during testing. The results obtained from the experiments conducted, Appium met the 8 criteria of these examiners. The author uses Appium automatically in testing for Android applications.

Key words:

Automation Testing, Appium, Android, Gherkin, Test Scenario

## KATA PENGANTAR

Puji syukur kita panjatkan kepada Tuhan Yang Maha Esa yang telah memberikan berkat dan karuniaNya kepada penulis selama pengerjaan Tugas Akhir ini dengan judul Implementasi Automation Test dengan Metode Black Box Testing pada Aplikasi Android menggunakan Appium dan Gherkin Syntax di PT. XYZ untuk memenuhi salah satu persyaratan dalam mencapai gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana. Penulis menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, proses pengerjaan Tugas Akhir ini tidak dapat terselesaikan dengan baik. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Kedua orangtua serta keluarga yang selalu memberikan dukungan doa dan semangat.
2. Ibu Harni Kusniyati, M.Kom selaku dosen pembimbing Tugas Akhir.
3. Bapak Hery Derajad Wijaya, S.Kom, MM selaku kaprodi Teknik Informatika Universitas Mercu Buana.
4. Bapak Wawan Gunawan, S.Kom., MT selaku Koordinator Tugas Akhir Teknik Informatika Universitas Mercu Buana.
5. Ibu Desi Ramayanti, S.Kom, MT selaku dosen pengampu mata kuliah Tugas Akhir Teknik Informatika Universitas Mercu Buana.
6. Serta pihak-pihak lainnya yang tidak dapat disebutkan satu per satu yang terlibat dan memberikan semangat dalam pengerjaan Tugas Akhir ini.

Akhir kata, penulis berharap Tugas Akhir ini dapat bermanfaat bagi pembaca. Penulis menyadari masih banyak kekurangan dikarenakan keterbatasan ilmu pengetahuan dan wawasan serta pengalaman yang penulis miliki. Penulis mohon maaf atas segala kekurangan tersebut dan berhasap adanya saran dan kritik yang membangun terhadap penelitian kedepannya. Terima kasih.

Jakarta, 3 Agustus 2021

Penulis

## DAFTAR ISI

HALAMAN SAMPUL.....	i
HALAMAN JUDUL .....	i
LEMBAR PERNYATAAN ORISINALITAS .....	ii
SURAT PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR... iii	
SURAT PERNYATAAN LUARAN TUGAS AKHIR.....	iv
LEMBAR PERSETUJUAN PENGUJI .....	v
LEMBAR PENGESAHAN .....	viii
ABSTRAK .....	ix
ABSTRACT.....	x
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xii
NASKAH JURNAL .....	1
KERTAS KERJA.....	2
BAB 1. LITERATUR REVIEW.....	3
BAB 2. ANALISIS DAN PERANCANGAN.....	60
BAB 3. SOURCE CODE.....	87
BAB 4. TAHAPAN EKSPERIMEN.....	141
BAB 5. HASIL SEMUA EKSPERIMEN.....	146
DAFTAR PUSTAKA .....	207
LAMPIRAN DOKUMEN HAKI.....	209
LAMPIRAN KORESPONDENSI .....	211

# IMPLEMENTASI AUTOMATION TEST DENGAN METODE BLACK BOX TESTING PADA APLIKASI ANDROID MENGGUNAKAN APPIUM DAN GHERKIN SYNTAX DI PT. XYZ

Louis Onike Munte<sup>1</sup>, Harni Kusniyati<sup>2</sup>

<sup>1,2</sup>Universitas Mercu Buana

Email: <sup>1</sup>41519110063@student.mercubuana.ac.id, <sup>2</sup>harni.kusniyati@mercubuana.ac.id

## Abstrak

*Testing* adalah proses mengevaluasi sistem atau komponen dengan maksud untuk menemukan bahwa apakah sistem tersebut memenuhi persyaratan yang ditentukan atau tidak. Saat ini, di PT. XYZ, pengujian aplikasi masih dilakukan secara manual. Hal ini membuat penguji cukup kewalahan dikarenakan dengan tersedianya sistem dalam beberapa layanan, yaitu web, *mobile web*, dan *mobile app*. Waktu yang dibutuhkan dalam proses pengujian lebih lama, sehingga proses rilis aplikasi juga menjadi lebih lama. Untuk itu perlu adanya peningkatan dalam proses pengujian aplikasi dengan mengotomatisasi pengujian. Tulisan ini membahas tentang implementasi pengujian otomatis pada aplikasi Android menggunakan Appium dengan format skenario uji menggunakan Gherkin *syntax*. Eksperimen dilakukan dengan menjalankan skenario uji sesuai dengan 8 kriteria pengujian yang ditetapkan yaitu waktu pengujian, bahasa pemrograman yang digunakan, dokumentasi hasil eksekusi kasus uji, dukungan proses *delay*, dukungan proses *suspend/resume*, dukungan proses *relaunch*, dukungan proses *scrolling*, serta penyimpanan perubahan selama pengujian. Hasil yang diperoleh dari eksperimen yang dilakukan, Appium memenuhi 8 kriteria pengujian tersebut. Penulis merekomendasikan penggunaan Appium dalam melakukan pengujian otomatis untuk aplikasi Android.

**Kata kunci:** *Pengujian Otomatis, Appium, Android, Gherkin, Skenario Uji*

## AUTOMATION TEST IMPLEMENTATION WITH BLACK BOX TESTING METHOD IN ANDROID APPLICATION USING APPIUM AND GHERKIN SYNTAX AT PT .XYZ

### Abstract

*Testing is an evaluation process for system or component with the intention of finding that whether the system meets the specified requirements or not. Currently, at PT. XYZ, application testing is still done manually. This made the testers quite overwhelmed because of the availability of the system in several services, namely web, mobile web and mobile applications. The time needed in the testing process is longer, so the application release process is also longer. For this reason, it is necessary to increase in the application testing process by automating testing. This paper discusses the implementation of automated testing on Android applications using Appium with a test scenario format using Gherkin syntax. Experiments were carried out by running scenario tests according to the 8 specified testing criteria, namely the time of the examiner, the programming language used, documentation of the results of the test case execution, support for the delay process, support for the suspension / resume process, support for the relaunch process, support for the scrolling process, and storage of changes during testing. The results obtained from the experiments conducted, Appium met the 8 criteria of these examiners. The author uses Appium automatically in testing for Android applications.*

**Keywords:** *Automation Testing, Appium, Android, Gherkin, Test Scenario*

### 1. PENDAHULUAN

Pada siklus pengembangan *software*, *testing* diperlukan untuk memastikan kualitas *software* sebelum dirilis (Min, J.L., Adila, I., Ani, R., 2020). Pemilihan teknik *testing* yang digunakan dalam pelaksanaan *testing* bergantung pada karakteristik *software* yang akan dirilis (Min, J.L., Adila, I., Ani, R., 2020). Dengan

adanya beberapa kegiatan yang dijalankan secara bersamaan, sangat penting untuk menjaga kinerja dari aplikasi tersebut agar tidak terjadi *error*. Proses untuk menemukan *error* pada aplikasi adalah proses *testing*. *Testing* adalah proses mengevaluasi sistem atau komponen dengan maksud untuk menemukan bahwa apakah sistem tersebut memenuhi persyaratan yang ditentukan atau tidak (Tutorialspoint, 2021). Pengujian

dilakukan untuk mengetahui efektivitas dan efisiensi *good test case* yang dipilih, guna memverifikasi aplikasi yang diuji sesuai dengan kebutuhan.

Proses pengujian aplikasi saat ini terbagi 2, yaitu *manual testing* dan *automation testing*. *Manual Testing* atau pengujian manual adalah langkah untuk mencari cacat atau *bug* pada program perangkat lunak, tanpa menggunakan bantuan dari *tools* atau *scripts* untuk memastikan jika aplikasi yang di uji bebas cacat dan aplikasi perangkat lunak dapat bekerja sesuai apa yang diharapkan. *Automation Testing* adalah langkah untuk mencari cacat atau *bug* pada program perangkat lunak dengan menggunakan bantuan dari *tools* atau *scripts* yang berjalan secara otomatis, untuk mengetahui apakah aplikasi berjalan sesuai dengan apa yang diharapkan (Ryanditha, P., 2018).

Keuntungan dilakukannya pengujian secara manual yaitu pengujian terhadap tampilan UI dari aplikasi dapat dilakukan, karena untuk pengujian secara otomatis tidak dapat dilakukan pengujian terhadap tampilan UI suatu aplikasi. Namun, untuk melakukan pengujian secara manual, terjadinya *human error* atau ketidaktekelitian sangat mungkin terjadi, terutama jika penguji sudah sangat familiar dengan aplikasi yang selalu di uji secara terus menerus. Jika penguji sudah sangat memahami alur dari aplikasi tersebut, penguji akan lebih mudah kelelahan sehingga dapat melewatkan beberapa hal dan membuat kesalahan. Jika menggunakan *automation testing* akan mengurangi *bug* yang terlewat dikarenakan *script* dijalankan secara otomatis. Dengan dilakukannya pengujian secara otomatis, pengujian dapat dilakukan lebih cepat dan efisien dalam menemukan *bug*.

Ada beberapa *tool* yang dapat digunakan untuk proses *automation testing*. Appium adalah *open-source tool* yang digunakan secara khusus untuk mengotomisasi aplikasi *mobile*. Dengan menggunakan Appium dapat memeriksa setiap elemen dalam aplikasi *mobile* untuk dapat mengidentifikasi elemen mana yang penting untuk diotomisasi dan bagaimana menghubungkan setiap elemen tersebut untuk membuat sistem otomisasi yang dapat bekerja dengan baik. Selain itu, Appium berfungsi sebagai *server* yang menghubungkan kode dari IntelliJ IDEA ke dalam aplikasi *mobile*, yang mewakili eksekusi *command* (Suharli, S., Dennis, A, 2018).

PT. XYZ adalah perusahaan yang bergerak dibidang layanan teknologi informasi yang menyediakan portal lowongan kerja. Yang menjadi objek pengujian pada penelitian ini adalah sistem lowongan kerja yang ada di PT. XYZ. Aplikasi ini diakses oleh pengguna melalui web dan juga *mobile web* serta *mobile app*. Untuk memastikan berbagai fitur pada aplikasi yang dibangun berjalan sesuai dengan kebutuhan, maka perlu dilakukan pengujian terhadap aplikasi tersebut sebelum dirilis ke pengguna. Kendala yang dialami oleh PT. XYZ saat ini adalah jadwal rilis sistem yang cukup lama. Hal

ini disebabkan oleh proses pengujian terhadap fitur-fitur sistem yang akan dirilis membutuhkan waktu yang lebih panjang. Saat ini, di PT. XYZ, pengujian aplikasi masih dilakukan secara manual. Hal ini membuat penguji cukup kewalahan dikarenakan dengan tersedianya sistem dalam beberapa layanan, yaitu web, *mobile web*, dan *mobile app*. Sementara pengujian terhadap sistem hanya dilakukan oleh satu orang penguji. Waktu yang dibutuhkan dalam proses pengujian lebih lama, sehingga proses rilis aplikasi juga menjadi lebih lama. Untuk itu perlu adanya peningkatan dalam proses pengujian aplikasi.

Pada penelitian ini dilakukan implementasi pengujian dengan metode *black box* terhadap aplikasi Android dengan menggunakan Appium di PT. XYZ. Dengan hasil akhir penerapan *automation testing* dengan gherkin *syntax* pada aplikasi Android di PT. XYZ. Oleh karena itu, topik ini penting dibahas supaya dapat diterapkan oleh perusahaan lainnya dalam melakukan pengujian aplikasi android.

## 2. METODE PENELITIAN

Metode penelitian yang diterapkan dalam pengerjaan penelitian ini yaitu *experimental research* dan *evaluative research*. *Experimental research* adalah metode penelitian kuantitatif yang digunakan untuk mengetahui pengaruh variabel independen (treatment/perlakuan) terhadap variabel dependen (hasil) dalam kondisi yang terkendali (Research, R., 2021). *Evaluative research* adalah kegiatan penelitian yang sifatnya mengevaluasi suatu kegiatan/program yang bertujuan untuk mengukur keberhasilan suatu kegiatan/program dan menentukan keberhasilan suatu program dan apakah telah sesuai dengan yang diharapkan (Kantun, S., 2017).

Pada penelitian ini, metode *experimental research* digunakan untuk membuktikan kemampuan *tool* Appium dalam melakukan pengujian aplikasi Android. Sedangkan metode *evaluative research* diterapkan saat melakukan analisis terhadap hasil pengujian dengan membandingkan hasil pengujian sesuai dengan tinjauan pustaka. Tahapan yang dilakukan yaitu sebagai berikut.

1. Pengumpulan Data  
Pada tahap ini penulis akan melakukan proses pengumpulan data dengan metode studi literatur dan observasi.
2. Rancangan Eksperimen  
Pada tahap ini penulis akan membuat rancangan eksperimen yang akan dilakukan yaitu dalam bentuk *test cases* dan *test scripts* yang akan dipakai dalam tahap eksperimen.
3. Eksperimen  
Pada tahap ini akan dilakukan implementasi eksperimen sesuai dengan rancangan eksperimen yang telah dibuat yang kemudian ditampilkan hasilnya.

## 2.1. Teori Pendukung

### a. Software Testing

Menurut IEEE Std 610.12-1990, *testing* didefinisikan sebagai proses sistem atau komponen yang beroperasi di bawah kondisi tertentu, mengamati atau merekam hasil, dan membuat evaluasi beberapa aspek dari sistem atau komponen. Semua fungsi-fungsi *software* harus diujicobakan, agar *software* bebas dari *error*, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya (Jatikusumo, D., Rahmat, R.H., 2019). Metodologi pengujian berdasarkan dibagi tiga yaitu *white box*, *grey box* dan *black box testing*.

#### 1. White Box Testing

*White box testing* merupakan pengujian yang didasarkan pada detail prosedur dan alur logika kode program suatu aplikasi. Pada kegiatan pengujian *white box*, tester melihat *source code* program untuk menemukan *bugs* dari kode program yang diuji. *White Box testing* adalah pengujian berdasarkan pengecekan terhadap detail perancangan, penggunaan struktur kontrol dari desain sistem secara prosedural untuk membagi pengujian ke dalam beberapa kasus pengujian yang dibuat dalam bentuk *flowchart* (Gunawan, W., 2019).

#### 2. Grey Box Testing

*Grey box testing* merupakan pengujian aplikasi dengan pengetahuan terbatas tentang cara kerja internal aplikasi. Dalam pengujian perangkat lunak, semakin banyak frasa yang diketahui, maka semakin memudahkan proses pengujian aplikasi. Pada *grey box testing*, penguji memiliki akses ke dokumen desain dan *database*. Dengan bekal tersebut, seorang penguji dapat menyiapkan data pengujian yang lebih baik dan menguji skenario saat membuat rencana pengujian.

#### 3. Black Box Testing

*Black-box testing* berfokus pada persyaratan fungsional perangkat lunak. Pengujian *black box* memungkinkan perekrutan perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.

### b. Automation Testing

*Automated testing* adalah proses membuat sebuah program (test script) yang mensimulasikan langkah-langkah *test case* manual dalam Bahasa pemrograman apapun dengan bantuan *external automation helper tool* lainnya. Eksekusi pengujian menggunakan *automated testing* membutuhkan konversi *test case* menjadi *test script*, yang setelah itu dilakukan *running test script* pada *automation tools* tersebut (Herlinda, Dona, K. & Erlin, W.A., 2019).

### c. Appium

Appium adalah *open-source tool* yang digunakan secara khusus untuk mengotomisasi aplikasi *mobile*. Dengan menggunakan Appium dapat memeriksa setiap elemen dalam aplikasi *mobile* untuk dapat mengidentifikasi elemen mana yang penting untuk diotomisasi dan bagaimana menghubungkan setiap elemen tersebut untuk membuat sistem otomisasi yang dapat bekerja dengan baik. Selain itu, Appium berfungsi sebagai *server* yang menghubungkan kode dari IntelliJ IDEA ke dalam aplikasi *mobile*, yang mewakili eksekusi *command* (Suharli, S., Dennis, A, 2018).

Appium adalah 'Server HTTP' yang ditulis menggunakan platform Node.js dan menjalankan sesi iOS dan Android menggunakan protokol penghubung JSON Webdriver. Oleh karena itu, sebelum menginisialisasi Appium Server, Node.js harus sudah diinstal sebelumnya disistem. Ketika Appium diunduh dan diinstal, maka server disiapkan di mesin yang mengekspos REST API. Maka akan menerima koneksi dan permintaan perintah dari klien dan menjalankan perintah tersebut pada perangkat seluler (Android / iOS). Appium akan menanggapi kembali dengan respon HTTP. Untuk menjalankan permintaan, Appium akan menggunakan *framework* otomisasi pengujian seluler untuk menampilkan antarmuka aplikasi. Framework yang digunakan:

- Instrumen Apple untuk iOS (Instrumen hanya tersedia di Xcode 3.0 atau lebih baru dengan OS X v10.5 dan lebih baru).
- Google UIAutomator untuk Android API level 16 atau lebih tinggi.
- Selendroid untuk Android API level 15 atau dibawahnya (Guru99, 2020a).

### d. Gherkin Syntax

Gherkin adalah bahasa yang dapat dibaca bisnis yang membantu untuk menggambarkan perilaku bisnis tanpa membahas detail implementasinya. Gherkin adalah bahasa khusus domain untuk menentukan pengujian dalam format spesifikasi Cucumber. Gherkin menggunakan bahasa sederhana untuk mendeskripsikan kasus penggunaan dan memungkinkan pengguna untuk menghapus detail logika dari uji perilaku.

Gherkin adalah bahasa berorientasi garis seperti YAML dan Python. Setiap baris disebut langkah, dimulai dengan kata kunci dan ujung terminal dengan tanda berhenti. Tab atau spasi digunakan untuk lekukan (Guru99, 2020b).

### e. Android

Android adalah sebuah sistem operasi atau *Operation System* (OS) berbasis Linux yang dirancang khusus untuk dipasang pada perangkat *smartphone* dan juga tablet, dan juga menyesuaikan dari

spesifikasi *low-end* hingga ke spesifikasi *high-end* (Abubakar, Z.F., 2019). Android memiliki sistem yang terbuka sehingga dapat diakses dan dikembangkan oleh semua orang dengan mudah sehingga Android menjadi salah satu sistem operasi yang banyak digunakan saat ini. Pengembang memiliki berbagai pilihan untuk membuat aplikasi berbasis Android.

### 3. RANCANGAN DAN EKSPERIMEN

Tabel 1 dibawah ini menjelaskan rancangan eksperimen yaitu skenario uji terhadap fitur yang dijalankan pada penelitian ini. Skenario uji akan dimuat dalam bentuk Gherkin *syntax* untuk memudahkan pengguna dalam memahami alur skenario uji.

**Tabel 1. Fitur dan Skenario Pengujian**

Fitur	Skenario
Register	<p>Verifikasi pengguna register dengan mengisi seluruh kolom dengan benar</p> <p>Verifikasi pengguna register dengan format email yang salah</p> <p>Verifikasi pengguna register dengan email yang sudah terdaftar</p> <p>Verifikasi pengguna register dengan kata sandi konfirmasi yang tidak sesuai</p> <p>Verifikasi pengguna register dengan mengisi kolom kata sandi kurang dari 6 karakter</p> <p>Verifikasi pengguna register dengan mengisi kolom kata sandi konfirmasi kurang dari 6 karakter</p>
Login	<p>Verifikasi pengguna login dengan email dan kata sandi yang sesuai</p> <p>Verifikasi pengguna login dengan format email yang salah</p> <p>Verifikasi pengguna login dengan kata sandi yang salah</p> <p>Verifikasi pengguna klik ikon LinkedIn pada halaman login</p> <p>Verifikasi pengguna klik ikon Google pada halaman login</p> <p>Verifikasi pengguna klik lupa kata sandi pada halaman login</p> <p>Verifikasi pengguna memasukkan email yang sesuai pada halaman lupa kata sandi</p> <p>Verifikasi pengguna memasukkan email yang salah pada halaman lupa kata sandi</p> <p>Verifikasi pengguna klik tombol 'Coba Lagi' pada pop up berhasil lupa kata sandi</p> <p>Verifikasi pengguna klik tombol 'Coba Lagi' pada pop up gagal lupa kata sandi</p> <p>Verifikasi pengguna klik 'Masuk' pada lupa kata sandi</p> <p>Verifikasi pengguna klik tombol 'Tidak' pada pop up gagal lupa kata sandi</p>
Beranda	<p>Verifikasi pengguna klik ikon notifikasi</p> <p>Verifikasi pengguna klik kolom pencarian</p> <p>Verifikasi pengguna klik tombol 'Tes Kemampuan'</p> <p>Verifikasi pengguna klik tombol 'Jadwal Interview'</p> <p>Verifikasi pengguna klik tombol 'Tes Online'</p> <p>Verifikasi pengguna menggeser item sesi kategori populer</p> <p>Verifikasi pengguna memilih salah satu item kategori populer</p> <p>Verifikasi pengguna menggeser ke kiri dan kanan sesi rekomendasi</p>
Status	<p>Verifikasi pengguna klik 'Lihat Semua' pada sesi rekomendasi</p> <p>Verifikasi pengguna menggeser ke kiri dan kanan sesi perusahaan</p> <p>Verifikasi pengguna memilih salah satu perusahaan</p> <p>Verifikasi pengguna klik 'Lihat Semua' pada sesi perusahaan</p> <p>Verifikasi pengguna menggeser ke kiri dan kanan sesi karir inspirasi</p> <p>Verifikasi pengguna memilih salah satu karir inspirasi</p> <p>Verifikasi pengguna klik 'Lihat Semua' pada sesi karir inspirasi</p> <p>Verifikasi pengguna klik menu 'Status'</p> <p>Verifikasi pengguna klik menu 'Lowongan'</p> <p>Verifikasi pengguna klik menu 'Disimpan'</p> <p>Verifikasi pengguna klik menu 'Profil'</p> <p>Verifikasi pengguna klik ikon kembali pada halaman pencarian</p> <p>Verifikasi pengguna memilih pencarian populer pada halaman pencarian</p> <p>Verifikasi pengguna klik ikon notifikasi</p> <p>Verifikasi pengguna klik tab 'Pribadi' dan 'Perusahaan'</p> <p>Verifikasi pengguna klik tombol 'Cari Pekerjaan' pada tab 'Pribadi'</p> <p>Verifikasi pengguna klik tombol 'Cari Perusahaan' pada tab 'Perusahaan'</p> <p>Verifikasi pengguna klik 'Lihat Semua' pada daftar status di tab 'Pribadi'</p> <p>Verifikasi pengguna klik 'Lihat Semua' pada daftar status di tab 'Perusahaan'</p> <p>Verifikasi pengguna klik data status di tab 'Pribadi'</p> <p>Verifikasi pengguna klik data status di tab 'Perusahaan'</p>
Lowongan	<p>Verifikasi pengguna mencari data lowongan</p> <p>Verifikasi pengguna klik ikon filter pada tab lowongan</p> <p>Verifikasi pengguna menyimpan lowongan</p> <p>Verifikasi pengguna klik tombol lamar lowongan</p> <p>Verifikasi pengguna membuka detail lowongan</p> <p>Verifikasi pengguna menyimpan dan membagi detail lowongan</p> <p>Verifikasi pengguna klik tombol lamar pada detail lowongan</p> <p>Verifikasi pengguna mencari data perusahaan</p> <p>Verifikasi pengguna klik ikon filter pada tab perusahaan</p> <p>Verifikasi pengguna klik tombol 'Subscribe' perusahaan</p> <p>Verifikasi pengguna membuka detail perusahaan</p> <p>Verifikasi pengguna membuka detail lowongan pada detail perusahaan</p>
Disimpan	<p>Verifikasi pengguna klik icon notifikasi</p> <p>Verifikasi pengguna klik tab 'Lowongan' dan tab 'Subscribe'</p> <p>Verifikasi pengguna klik tombol 'Cari Pekerjaan' di tab 'Lowongan'</p> <p>Verifikasi pengguna klik tombol 'Cari Perusahaan' di tab 'Subscribe'</p> <p>Verifikasi pengguna membuka detail lowongan</p> <p>Verifikasi pengguna membatalkan simpan lowongan</p> <p>Verifikasi pengguna membatalkan subscribe perusahaan</p>

Fitur	Skenario
Jadwal Interview	Verifikasi pengguna klik ikon kembali Verifikasi pengguna klik tab 'Interview' dan tab 'Video Interview' Verifikasi pengguna klik tombol 'Cari Pekerjaan' pada tab 'Interview' Verifikasi pengguna klik tombol 'Cari Pekerjaan' pada tab 'Video Interview'
Profil	Verifikasi pengguna klik ikon notifikasi Verifikasi pengguna klik ikon pengaturan Verifikasi pengguna klik submenu profil Verifikasi pengguna klik submenu pengalaman Verifikasi pengguna klik submenu Pendidikan Verifikasi pengguna klik submenu Organisasi Verifikasi pengguna klik submenu Kemampuan dan Bahasa Verifikasi pengguna klik submenu CV dan Dokumen Verifikasi pengguna klik submenu Sertifikat Verifikasi pengguna klik submenu Hasil Tes Kemampuan Verifikasi pengguna klik submenu Preferensi
Profil Organisasi	Verifikasi pengguna menambah data organisasi Verifikasi pengguna mengubah data organisasi Verifikasi pengguna menghapus data organisasi
Profil Kemampuan dan Bahasa	Verifikasi pengguna menambah data kemampuan Verifikasi pengguna mengubah data kemampuan Verifikasi pengguna menghapus data kemampuan Verifikasi pengguna menambah data bahasa Verifikasi pengguna mengubah data bahasa Verifikasi pengguna menghapus data bahasa
Profil CV dan Dokumen	Verifikasi pengguna menambah data CV Verifikasi pengguna mengunduh data CV Verifikasi pengguna menambah data ijazah Verifikasi pengguna mengunduh data ijazah Verifikasi pengguna menambah data KTP Verifikasi pengguna mengunduh data KTP
Profil Sertifikat	Verifikasi pengguna menambah data sertifikat Verifikasi pengguna mengubah data sertifikat Verifikasi pengguna menghapus data sertifikat
Profil Preferensi	Verifikasi pengguna memilih opsi 'Ya' dan 'Tidak' pada halaman preferensi Verifikasi pengguna klik kolom posisi Verifikasi pengguna klik kolom lokasi Verifikasi pengguna mengubah data preferensi posisi Verifikasi pengguna mengubah data preferensi lokasi Verifikasi pengguna mengubah data preferensi gaji
Profil Pengaturan	Verifikasi pengguna mengubah bahasa ke bahasa Indonesia Verifikasi pengguna klik ikon pengaturan Verifikasi pengguna mengubah bahasa ke bahasa Inggris Verifikasi pengguna mengubah kata sandi dengan data yang benar Verifikasi pengguna mengubah kata sandi dengan konfirmasi sandi yang tidak sesuai Verifikasi pengguna mengubah kata sandi dengan kata sandi saat ini yang tidak sesuai Verifikasi pengguna klik 'Syarat Ketentuan'

Fitur	Skenario
	Verifikasi pengguna klik 'Tentang Aplikasi' Verifikasi pengguna klik 'Pusat Bantuan' Verifikasi pengguna klik 'Logout'

Tabel 2 dibawah ini menjelaskan kriteria-kriteria pengujian yang dibahas pada penelitian sebelumnya (Barus, Leo, 2019). Kriteria pengujian ini dijadikan dasar pemenuhan fungsionalitas aplikasi pengujian otomatisasi pada aplikasi android yang diuji.

**Tabel 2 Kriteria Pengujian**

No.	Kriteria
1.	Waktu yang dibutuhkan untuk mengeksekusi kasus uji
2.	Bahasa pemrograman yang digunakan
3.	Dokumentasi hasil eksekusi kasus uji dari alat pengujian
4.	Apakah alat pengujian mendukung penundaan ( <i>delay</i> ) selama melakukan pengujian
5.	Apakah alat pengujian mendukung pengangguhan atau mulai ulang ( <i>Suspend/Resume</i> ) aplikasi saat melakukan pengujian
6.	Apakah alat pengujian dapat menjalankan ulang ( <i>re-launch</i> ) aplikasi selama melakukan pengujian
7.	Apakah alat pengujian dapat menampilkan bagian-bagian yang tidak muat di layer ( <i>scrolling</i> ) saat melakukan pengujian
8.	Apakah perubahan yang dilakukan selama melakukan pengujian menggunakan alat pengujian tersimpan

Kriteria 1 dilakukan untuk mengetahui waktu yang dibutuhkan dalam menjalankan skenario uji. Kriteria ini akan diuji pada seluruh skenario fitur. Kriteria 2 dilakukan untuk mengetahui bahasa pemrograman yang dapat digunakan pada Appium. Kriteria ini akan diuji pada fungsi yang akan dihasilkan untuk menjalankan skenario otomatisasi. Jika eksekusi pengujian telah selesai dilakukan, diperlukan dokumentasi hasil uji. Untuk itu Kriteria 3 dilakukan untuk mengetahui laporan hasil pengujian yang dijalankan menggunakan Appium. Kriteria ini akan diuji pada seluruh skenario fitur. Ketika menjalankan skenario uji, diperlukan kondisi menunda menjalankan skenario untuk menunggu hasil dari langkah yang berjalan tampil. Kriteria 4 dilakukan untuk memenuhi kondisi tersebut. Kriteria ini dijalankan pada skenario fitur login dan register. Kriteria 5 dilakukan untuk mengetahui jika dilakukan penangguhan saat pengujian berjalan, kondisi aplikasi terakhir akan tersimpan atau tidak. Kriteria ini akan diuji pada skenario fitur profil sertifikat dan profil preferensi. Kriteria 6 dilakukan untuk mengetahui apakah Appium dapat melakukan re-launch aplikasi ketika menjalankan skenario. Kriteria ini diuji pada semua fitur. Kriteria 7 dilakukan untuk mengetahui apakah Appium dapat melakukan *scrolling* halaman aplikasi pada proses pengujian. Kriteria ini akan diuji pada skenario fitur beranda dan fitur profil. Kriteria 8 dilakukan untuk mengetahui apabila saat menjalankan skenario uji terdapat perubahan, perubahan tersebut akan tersimpan atau tidak. Kriteria ini akan diuji pada skenario fitur profil organisasi dan profil kemampuan dan bahasa.

#### 4. HASIL DAN PEMBAHASAN

Hasil eksperimen untuk kriteria pengujian waktu yang dibutuhkan untuk mengeksekusi kasus uji dapat dilihat di Tabel 3.

Tabel 3 Hasil Pengujian dan Waktu Eksekusi

No.	Fitur	Status Uji Skenario dan Langkah	Waktu Eksekusi
1.	Register	6 skenario (6 lulus) 66 langkah (66 lulus)	25 menit 14,778 detik
2.	Login	12 skenario (12 lulus) 60 langkah (60 lulus)	40 menit 26,074 detik
3.	Beranda	21 skenario (1 gagal, 20 lulus) 92 langkah (1 gagal, 91 lulus)	115 menit 53,448 detik
4.	Status	8 skenario (8 lulus) 42 langkah (42 lulus)	42 menit 4,070 detik
5.	Lowongan dan Perusahaan	7 skenario (7 lulus) 44 langkah (44 lulus) 5 skenario (5 lulus) 37 langkah (37 lulus)	38 menit 53,864 detik 26 menit 14,967 detik
6.	Disimpan	7 skenario (7 lulus) 34 langkah (34 lulus)	33 menit 59,543 detik
7.	Jadwal Interview	4 skenario (4 lulus) 18 langkah (18 lulus)	28 menit 55,214 detik
8.	Profil	11 skenario (11 lulus) 55 langkah (55 lulus)	63 menit 42,302 detik
9.	Profil Organisasi	3 skenario (3 lulus) 35 langkah (35 lulus)	22 menit 14,634 detik
10.	Profil Kemampuan dan Bahasan	6 skenario (6 lulus) 57 langkah (57 lulus)	41 menit 22,668 detik
11.	Profil CV dan Dokumen	6 skenario (6 lulus) 39 langkah (39 lulus)	38 menit 0,827 detik
12.	Profil Sertifikat	3 skenario (3 lulus) 30 langkah (30 lulus)	21 menit 13,982 detik
13.	Profil Preferensi	6 skenario (6 lulus) 35 langkah (35 lulus)	36 menit 3,374 detik
14.	Profil Pengaturan	10 skenario (10 lulus) 87 langkah (87 lulus)	60 menit 34,242 detik

Untuk mengeksekusi 116 skenario dengan 731 langkah dibutuhkan waktu 10 jam 34 menit 0,8998 detik.

Appium dapat digunakan dengan beberapa bahasa pemrograman, namun hasil dari eksperimen untuk kriteria pengujian bahasa pemrograman yang digunakan adalah bahasa pemrograman Ruby.

Hasil dari eksperimen untuk kriteria pengujian dokumentasi hasil eksekusi kasus uji dari alat pengujian adalah laporan hasil pengujian dengan format HTML dan *command line*.

Hasil dari eksperimen untuk kriteria pengujian apakah alat pengujian mendukung penundaan (*delay*) selama melakukan pengujian adalah Appium mendukung keadaan *delay* selama menjalankan pengujian pada fitur login dan register. Proses penundaan dilakukan dengan menambahkan fungsi *user wait* untuk menunggu aplikasi yang diuji memuat tampilan halaman aplikasi secara utuh.

Hasil dari eksperimen untuk kriteria pengujian apakah alat pengujian mendukung pengangguhan (*Suspend/Resume*) adalah Appium mendukung keadaan *suspend/resume* selama melakukan pengujian pada fitur profil sertifikat dan profil preferensi. Ketika proses pengujian sedang dijalankan, penguji melakukan *suspend* dengan menekan tombol *home* pada emulator, kemudian membuka kembali aplikasi yang diuji. Aplikasi menampilkan kondisi yang sama dengan kondisi sesaat sebelum di *suspend* dan kemudian proses pengujian tetap dilanjutkan.

Hasil dari eksperimen untuk kriteria pengujian apakah alat pengujian dapat menjalankan ulang (*re-launch*) aplikasi selama melakukan pengujian adalah Appium mendukung proses menjalankan ulang aplikasi. Ketika skenario baru dijalankan pada Appium, maka aplikasi yang diuji akan dimulai kembali dari awal.

Hasil dari eksperimen untuk kriteria pengujian apakah alat pengujian dapat menampilkan bagian-bagian yang tidak muat di layer (*scrolling*) saat melakukan pengujian adalah Appium dapat melakukan *scrolling* ketika menjalankan pengujian pada fitur beranda dan profil. Proses *scrolling* dijalankan menggunakan fungsi *scrolling* hingga menemukan kata kunci yang telah ditetapkan pada masing-masing skenario.

Hasil dari eksperimen untuk kriteria pengujian apakah perubahan yang dilakukan selama melakukan pengujian menggunakan alat pengujian tersimpan adalah Appium menyimpan perubahan yang terjadi selama proses pengujian pada fitur profil organisasi serta profil kemampuan dan bahasa dijalankan. Skenario yang dijalankan adalah proses menambah data, merubah data dan menghapus data.

Berdasarkan hasil eksperimen yang telah dilakukan pada aplikasi uji menggunakan Appium, maka diketahui bahwa Appium memenuhi delapan kriteria pengujian yang ditetapkan. Proses pengujian menggunakan Appium pada aplikasi uji pada penelitian

ditemukan bahwa rata-rata waktu yang dibutuhkan untuk menjalankan 1 langkah uji adalah 52,62 detik. Appium dapat dijalankan menggunakan bahasa pemrograman Ruby. Dokumentasi hasil pengujian menggunakan Appium disediakan dalam format HTML dan *command line*. Appium mendukung proses penundaan (*delay*), pengangguhan (*suspend/resume*), menjalankan ulang (*re-launch*), serta *scrolling*. Appium juga menyimpan proses perubahan yang dilakukan selama proses pengujian.

## 5. KESIMPULAN DAN SARAN

Alat pengujian yang digunakan yaitu Appium memenuhi seluruh kriteria pengujian yang ditetapkan. Proses penulisan kasus uji dilakukan menggunakan Gherkin *syntax* untuk memudahkan pengguna dalam memahami alur skenario uji.

Rata-rata waktu yang dibutuhkan untuk menjalankan 1 langkah uji pada Appium adalah 52,62 detik. Bahasa pemrograman yang digunakan adalah Ruby. Dokumentasi hasil pengujian menggunakan Appium disediakan dalam format HTML dan *command line*. Proses penundaan (*delay*), pengangguhan (*suspend/resume*), menjalankan ulang (*re-launch*), serta *scrolling* didukung pada penggunaan Appium. Dan proses perubahan yang dilakukan selama proses pengujian juga disimpan ketika menggunakan Appium.

Penelitian ini merekomendasikan penggunaan Appium dalam proses pengujian aplikasi *mobile* berbasis Android.

Dalam penelitian lanjutan disarankan aplikasi uji yang digunakan sudah menyediakan *identifier* pada setiap elemen-elemen yang ada, agar lebih memudahkan dalam proses penentuan *selector* yang akan digunakan dalam menjalankan pengujian.

## DAFTAR PUSTAKA

- ABUBAKAR, Z.F., 2019. Android (Sistem Operasi): Pengertian, Sejarah dan Nama Versi. Tersedia di: <<https://tekno.foresteract.com/android/>> [Diakses 4 Juni 2021]
- BARUS, A.C., LEO, S., 2019. Studi Perbandingan Alat Pengujian Otomatis Untuk Aplikasi Android. Tersedia di: <<https://jtiik.ub.ac.id/index.php/jtiik/article/view/953>> [Diakses 17 Mei 2021]
- GUNAWAN, W., 2019. Pengembangan Aplikasi Berbasis Android Untuk Pengenalan Huruf Hijaiah. Tersedia di: <<https://ejournal.bsi.ac.id/ejurnal/index.php/ji/article/view/5373>> [Diakses 17 Mei 2021]
- GURU99. Appium Tutorial for Android & IOS Mobile Apps Testing. Tersedia di: <<https://www.guru99.com/introduction-to-appium.html>> [Diakses 17 Mei 2021]
- GURU99. Gherkin Language: Format, Syntax & Gherkin Test in Cucumber. Tersedia di: <<https://www.guru99.com/gherkin-test-cucumber.html>> [Diakses 17 Mei 2021]
- HERLINDA., DONA, K. & ERLIN, W.A., 2019. Automation Testing Tool Dalam Pengujian Aplikasi Belajar Tajwid Pada Platform Android. Tersedia di: <<https://journal.lppmunindra.ac.id/index.php/STRING/article/view/5285>> [Diakses 17 Mei 2021]
- JATIKUSUMO, D., RAHMAT, R.H., 2019. Pembelajaran Bahasa Inggris Melalui Permainan Tebak Kata dengan Web Crawler Menggunakan Android. Tersedia di: <<https://journal.untar.ac.id/index.php/computatio/article/view/3366>> [Diakses 17 Mei 2021]
- KANTUN, S., 2017. Penelitian Evaluatif sebagai Salah Satu Model Penelitian Dalam Bidang Pendidikan. Tersedia di: <<https://jurnal.unej.ac.id/index.php/JPE/article/view/3809>> [Diakses 4 Juni 2021]
- MIN, J.L., ADILA, I., ANI, R., 2020. Evaluasi Penggunaan Manual dan Automated Software Testing pada Pelaksanaan End-to-End Testing. Tersedia di: <<https://jurnal.polindra.ac.id/index.php/jtt/article/view/256>> [Diakses 4 Juni 2021]
- RESEARCH, R., 2021. Pengertian Metode Penelitian Eksperimen dan Cara Menggunakannya. Tersedia di: <<https://ranahresearch.com/pengertian-metode-penelitian-eksperimen/>> [Diakses 4 Juni 2021]
- RYANDITHA, P., 2018. Perbedaan Manual Testing dan Automated Testing. Tersedia di: <<https://medium.com/skyshidigital/perbedaan-manual-testing-dan-automated-testing-9d13373a36e>> [Diakses 17 Mei 2021]
- SUHARLI, S., DENNIS A., 2018. Automation Testing Tools. Tersedia di: <<https://socs.binus.ac.id/2018/12/21/automation-testing-tools>> [Diakses 17 Mei 2021]
- TUTORIALSPPOINT., 2021. Software Testing Tutorial. Tersedia di: <[http://www.tutorialspoint.com/software\\_testing](http://www.tutorialspoint.com/software_testing)> [Diakses 17 Mei 2021]

## KERTAS KERJA

### Ringkasan

Kertas kerja ini merupakan material kelengkapan artikel jurnal dengan judul Implementasi Automation Test dengan Metode Black Box Testing pada Aplikasi Android menggunakan Appium dan Gherkin Syntax di PT. XYZ. Kertas kerja berisi semua material hasil penelitian Tugas Akhir yang tidak dimuat/atau disertakan di artikel jurnal. Di dalam kertas kerja ini disajikan: literature review, source code, dan hasil eksperimen secara keseluruhan.

