

**ANALISA PERBANDINGAN *BLACK-BOX*
AUTOMATED TESTING DAN *MANUAL TESTING*
PADA APLIKASI ACCMART**

Tugas Akhir

**Diajukan untuk Memenuhi Salah Satu Persyaratan Mencapai Derajat
Sarjana Informatika**



Dibuat Oleh:

HARTONO RUSLI

150708608

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ATMA JAYA YOGYAKARTA
2020**

HALAMAN PENGESAHAN

Tugas Akhir Berjudul

ANALISA PERBANDINGAN AUTOMATED TESTING DAN MANUAL TESTING PADA APLIKASI
ACCMART

yang disusun oleh

HARTONO RUSLI

150708608

dinyatakan telah memenuhi syarat pada tanggal 03 Juni 2020

		Keterangan
Dosen Pembimbing 1	: Martinus Maslim, ST., MT.	Telah menyetujui
Dosen Pembimbing 2	: Stephanie Pamela Adithama, ST., MT.	Telah menyetujui
Tim Penguji		
Penguji 1	: Martinus Maslim, ST., MT.	Telah menyetujui
Penguji 2	: Dr. Andi Wahyu Rahardjo, BSEE., MSSE	Telah menyetujui
Penguji 3	: Yulius Harjoseputro, ST., MT.	Telah menyetujui

Yogyakarta, 03 Juni 2020

Universitas Atma Jaya Yogyakarta

Fakultas Teknologi Industri

Dekan

ttd

Dr. A. Teguh Siswanto, M.Sc

PERNYATAAN ORISINALITAS & PUBLIKASI ILMIAH

Saya yang bertanda tangan di bawah ini:

Nama Lengkap : Hartono Rusli
NPM : 150708608
Program Studi : Informatika
Fakultas : Teknologi Industri
Judul Penelitian : Analisa Perbandingan *Black-Box Automated Testing* dan *Manual Testing* Pada Aplikasi ACCMart

Menyatakan dengan ini:

1. Tugas Akhir ini adalah benar tidak merupakan salinan sebagian atau keseluruhan dari karya penelitian lain.
2. Memberikan kepada Universitas Atma Jaya Yogyakarta atas penelitian ini, berupa Hak untuk menyimpan, mengelola, mendistribusikan, dan menampilkan hasil penelitian selama tetap mencantumkan nama penulis.
3. Bersedia menanggung secara pribadi segala bentuk tuntutan hukum atas pelanggaran Hak Cipta dalam pembuatan Tugas Akhir ini.

Demikianlah pernyataan ini dibuat dan dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 4 Mei 2020

Yang menyatakan,

Hartono Rusli

150708608

PERNYATAAN PERSETUJUAN DARI INSTANSI ASAL PENELITIAN

Saya yang bertanda tangan di bawah ini:

Nama Lengkap Pembimbing : Mutiara Caesagusta Yosadhie

Jabatan : *Quality Control Coordinator*

Departemen : *Department of IT Architecture and Governance*

Menyatakan dengan ini:

Nama Lengkap : Hartono Rusli

NPM : 150708608

Program Studi : Informatika

Fakultas : Teknologi Industri

Judul Penelitian : Analisa Perbandingan *Black-Box Automated Testing* dan *Manual Testing* Pada Aplikasi ACCMart

1. Penelitian telah selesai dilaksanakan pada perusahaan.
2. Perusahaan telah melakukan sidang internal berupa kelayakan penelitian ini dan akan mencantumkan lembar penilaian secara tertutup kepada pihak universitas sebagai bagian dari nilai akhir mahasiswa.
3. Memberikan kepada Instansi Penelitian dan Universitas Atma Jaya Yogyakarta atas penelitian ini, berupa hak untuk menyimpan, mengelola, mendistribusikan, dan menampilkan hasil penelitian selama tetap mencantumkan nama penulis.

Demikianlah pernyataan ini dibuat dan dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 4 Mei 2020

Yang menyatakan,

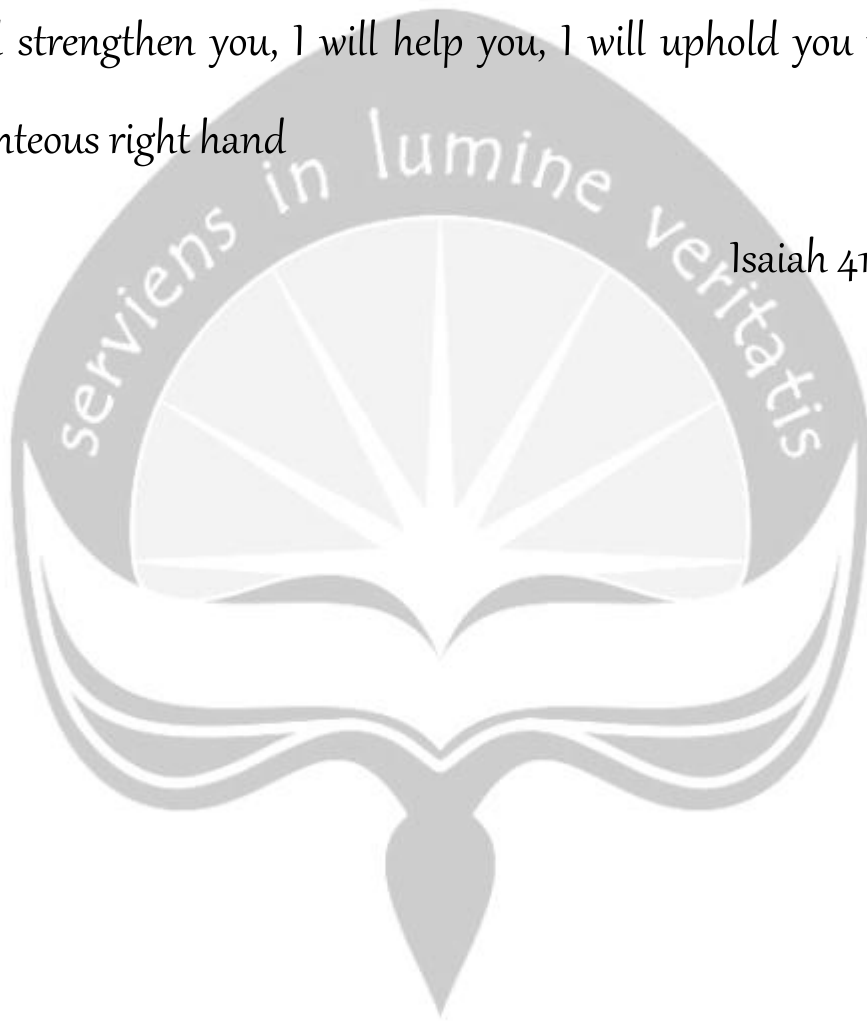
Mutiara Caesagusta Yosadhie

Department of IT Architecture and Governance

HALAMAN PERSEMBAHAN

Fear not, for I am with you; be not dismayed, for I am your God; I will strengthen you, I will help you, I will uphold you with my righteous right hand

Isaiah 41 : 10 ESV



KATA PENGANTAR

Puji dan syukur penulis haturkan kepada Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya penulis dapat menyelesaikan pembuatan tugas akhir “Analisa Perbandingan Black-Box Automated Testing dan Manual Testing Pada Aplikasi ACCMart” ini dengan baik.

Penulisan tugas akhir ini bertujuan untuk memenuhi salah satu syarat untuk mencapai derajat sarjana Teknik Informatika dari Program Studi Informatika, Fakultas Teknologi Industri di Universitas Atma Jaya Yogyakarta.

Penulis menyadari bahwa dalam pembuatan tugas akhir ini penulis telah mendapatkan bantuan, bimbingan, dan dorongan dari banyak pihak. Untuk itu, pada kesempatan ini penulis ingin mengucapkan terima kasih kepada:

1. Tuhan Yesus Kristus yang selalu membimbing dalam iman-Nya, memberikan berkat-Nya, dan menyertai penulis selalu.
2. Bapak Dr. A. Teguh Siswanto M.Sc., selaku Dekan Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta.
3. Bapak Martinus Maslim S.T., M.T., selaku dosen pembimbing I yang telah membimbing dan memberikan masukan serta motivasi kepada penulis untuk menyelesaikan tugas akhir ini.
4. Ibu Stephanie Pamela S.T., M.T., selaku dosen pembimbing II yang telah membimbing dan memberikan masukan serta motivasi kepada penulis untuk menyelesaikan tugas akhir ini.
5. Orang Tua dan saudara yang selalu *men-support* penulis sehingga bisa menyelesaikan kuliah.
6. Lun Zi Juan dan Wong Jing Yee sebagai teman terdekat dan telah menemani penulis selama beberapa tahun terakhir di Yogyakarta.
7. Kak Endah, Kak Aji, Ps. Jimmy dan keluarga, Aaron, Daniel, Wei Leng, Sam,

Ms Dita, Timmy, Ian, Hon, dan masih banyak lagi, serta keluarga besar EN Jogja yang telah menjadi teman dekat penulis.

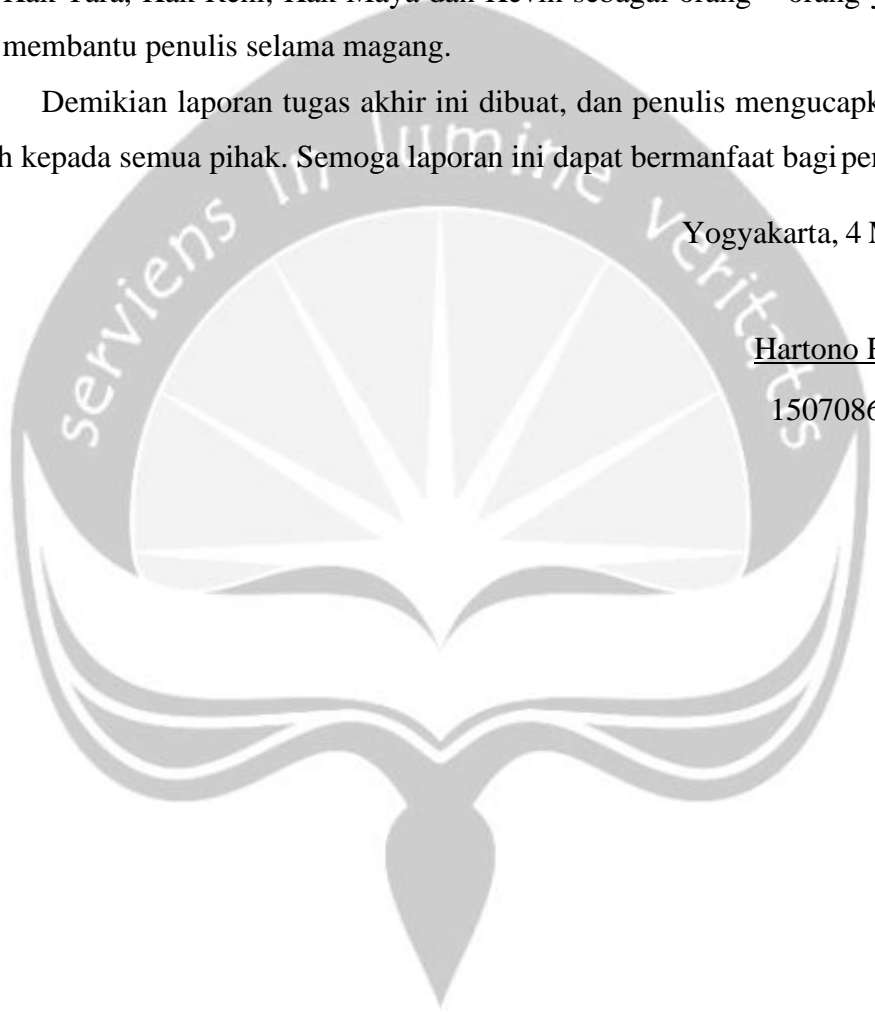
8. Teman – teman dari *batch* 3 & 4 selama magang.
9. Astra Credit Companies yang telah memberikan banyak pengalaman dan kesempatan untuk magang.
10. Kak Tara, Kak Reni, Kak Maya dan Kevin sebagai orang – orang yang telah membantu penulis selama magang.

Demikian laporan tugas akhir ini dibuat, dan penulis mengucapkan terima kasih kepada semua pihak. Semoga laporan ini dapat bermanfaat bagi pembaca.

Yogyakarta, 4 Mei 2020

Hartono Rusli

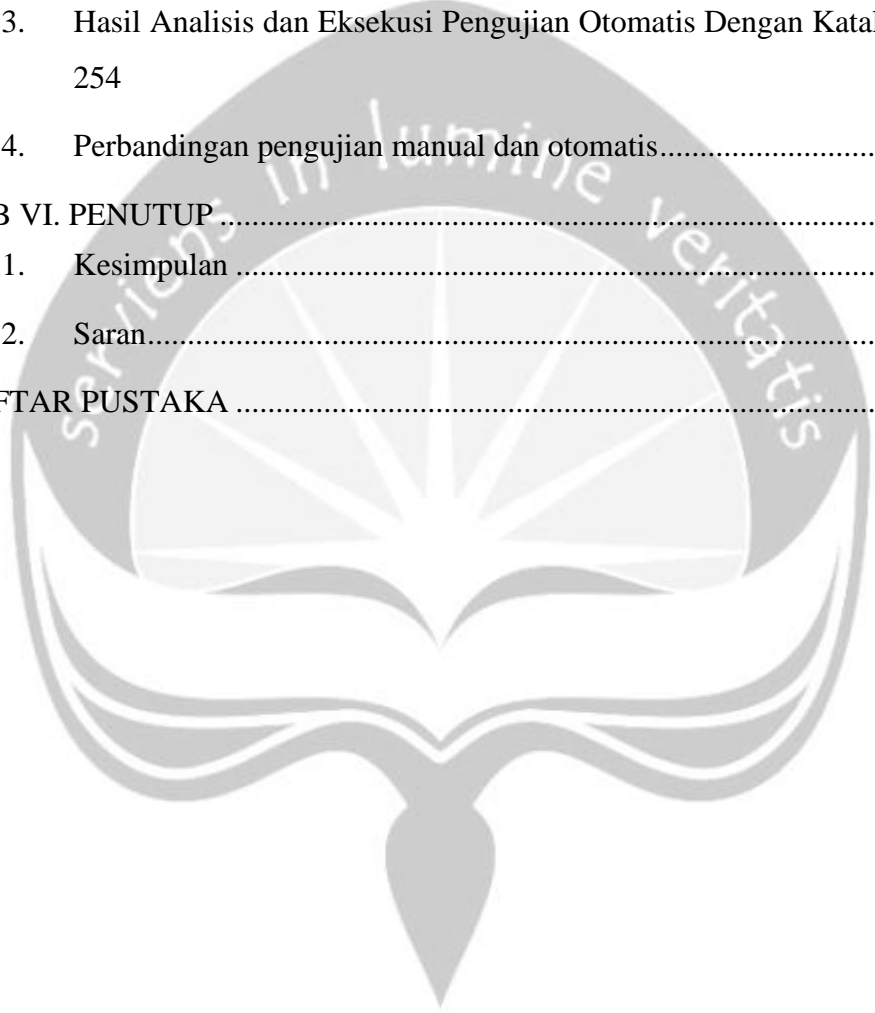
150708608



DAFTAR ISI

LEMBAR PENGESAHAN	Error! Bookmark not defined.
PERNYATAAN ORISINALITAS & PUBLIKASI ILMIAH.....	i
PERNYATAAN PERSETUJUAN DARI INSTANSI ASAL PENELITIAN	iii
HALAMAN PERSEMBAHAN	iv
KATA PENGANTAR	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL.....	xi
BAB I. PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian	3
1.5. Metode Penelitian.....	3
1.6. Sistematika Penulisan	4
BAB II. TINJAUAN PUSTAKA.....	6
BAB III. LANDASAN TEORI.....	11
3.1. ACCMart.....	11
3.2. <i>Software Bug</i>	11
3.3. <i>Software Testing</i>	12
3.4. Katalon Studio.....	13
3.5. <i>Black-Box Testing</i>	13
BAB IV. ANALISIS DAN PERANCANGAN EKSPERIMEN	15
4.1. Deskripsi Masalah.....	15
4.2. Analisis <i>Functional Flow</i> Pada Fungsi Yang Telah Ditentukan Dalam Penelitian.....	15

4.3.	Analisis dan Perancangan Skenario Pengujian	17
4.4.	Analisis Kebutuhan Alat	18
4.5.	Analisis dan Perancangan <i>Test Case</i>	19
BAB V. HASIL EKSPERIMEN DAN PEMBAHASAN		139
5.1.	Deskripsi Eksperimen	139
5.2.	Hasil Eksekusi Pengujian Manual.....	139
5.3.	Hasil Analisis dan Eksekusi Pengujian Otomatis Dengan Katalon Studio 254	
5.4.	Perbandingan pengujian manual dan otomatis.....	278
BAB VI. PENUTUP		284
6.1.	Kesimpulan	284
6.2.	Saran.....	285
DAFTAR PUSTAKA		286



DAFTAR GAMBAR

Gambar 1. Test case Login dalam Katalon Studio.....	254
Gambar 2. Test Suite Login Dalam Katalon Studio	254
Gambar 3. Variabel Dalam Katalon Untuk Test Case Login.....	255
Gambar 4. Data Binding Dalam Katalon Studio.....	255
Gambar 5. Data Microsoft Excel Yang Akan Dimasukkan Ke Katalon Studio ..	256
Gambar 6. Hasil Eksekusi Test Suite Login Dalam Katalon Studio	256
Gambar 7. Test Case Log Out Dalam Katalon Studio	257
Gambar 8. Hasil Eksekusi Test Case Log Out.....	257
Gambar 9. Test Case Ubah Kata Sandi Dalam Katalon Studio(1)	258
Gambar 10. Test Case Ubah Kata Sandi Dalam Katalon Studio(2)	258
Gambar 11. Test Case Ubah Kata Sandi Dalam Katalon Studio(3)	258
Gambar 12. Variabel Dalam Katalon Studio Untuk Test Case Ubah Kata Sandi	259
Gambar 13. Test Suite Dalam Katalon Studio Untuk Ubah Kata Sandi.....	259
Gambar 14. Data Binding Dalam Katalon Studio Untuk Ubah Kata Sandi	260
Gambar 15. Data Excel Untuk Ubah Kata Sandi.....	260
Gambar 16. Hasil Eksekusi Test Suite Ubah Kata Sandi	261
Gambar 17. Test Case Dalam Katalon Studio Untuk Hubungi Kami(1).....	261
Gambar 18. Test Case Dalam Katalon Studio Untuk Hubungi Kami(2).....	262
Gambar 19. Variabel Dalam Katalon Studio Untuk Hubungi Kami	262
Gambar 20. Test Suite Dalam Katalon Studio Untuk Hubungi Kami	263
Gambar 21. Data Binding Dalam Katalon Studio Untuk Hubungi Kami.....	263
Gambar 22. Data Excel Untuk Hubungi Kami	263
Gambar 23. Hasil Eksekusi Test Suite Hubungi Kami.....	264
Gambar 24. Test Case Dalam Katalon Studio Untuk Edit Profil(1).....	264
Gambar 25. Test Case Dalam Katalon Studio Untuk Edit Profil(2).....	265
Gambar 26. Test Case Dalam Katalon Studio Untuk Edit Profil(3).....	265
Gambar 27. Test Case Kalender Dalam Katalon Studio Untuk Edit Profil	265
Gambar 28. Variabel Dalam Katalon Studio Untuk Edit Profil.....	266
Gambar 29. Test Suite Dalam Katalon Studio Untuk Edit Profil	267

Gambar 30. Data Binding Dalam Katalon Studio Untuk Edit Profil.....	267
Gambar 31. Data Excel Untuk Edit Profil	267
Gambar 32. Hasil Eksekusi Test Suite Edit Profil(1)	268
Gambar 33. Hasil Eksekusi Test Suite Edit Profil(2)	268
Gambar 34. Hasil Eksekusi Test Suite Edit Profil(3)	268
Gambar 35. Hasil Eksekusi Test Suite Edit Profil(4)	269
Gambar 36. Hasil Eksekusi Test Suite Edit Profil	269
Gambar 37. Test Case Dalam Katalon Studio Untuk Kalkulator Kredit(1)	270
Gambar 38. Test Case Dalam Katalon Studio Untuk Kalkulator Kredit(2)	270
Gambar 39. Variabel Dalam Katalon Studio Untuk Kalkulator Kredit	271
Gambar 40. Test Suite Dalam Katalon Studio Untuk Kalkulator Kredit.....	272
Gambar 41. Data Binding Dalam Katalon Studio Untuk Kalkulator Kredit	272
Gambar 42. Data Excel Untuk Kalkulator Kredit.....	273
Gambar 43. Hasil Eksekusi Test Suite Kalkulator Kredit(1).....	273
Gambar 44. Hasil Eksekusi Test Suite Kalkulator Kredit(2).....	273
Gambar 45. Hasil Eksekusi Test Suite Kalkulator Kredit(3).....	274
Gambar 46. Hasil Eksekusi Test Suite Kalkulator Kredit(4).....	274
Gambar 47. Test Case Dalam Katalon Studio Untuk Kalkulator Budget(1)	275
Gambar 48. Test Case Dalam Katalon Studio Untuk Kalkulator Budget(2)	275
Gambar 49. Variabel Dalam Katalon Studio Untuk Kalkulator Budget.....	276
Gambar 50. Test Suite Dalam Katalon Studio Untuk Kalkulator Budget	276
Gambar 51. Data Binding Dalam Katalon Studio Untuk Kalkulator Budget	277
Gambar 52. Data Excel Untuk Kalkulator Budget	277
Gambar 53. Hasil Eksekusi Test Suite Kalkulator Budget(1)	277
Gambar 54. Hasil Eksekusi Test Suite Kalkulator Budget(2)	277
Gambar 55. Hasil Eksekusi Test Suite Kalkulator Budget(3)	278

DAFTAR TABEL

Tabel 1. Perbandingan Penelitian Dengan Penelitian Sebelumnya	10
Tabel 2. Functional Flow Fungsi Dalam Penelitian.....	15
Tabel 3.Skenario Pengujian Fungsi	17
Tabel 4. Rancangan Pengujian Daftar – Alur Utama Positif	19
Tabel 5. Rancangan Pengujian Daftar – Inputan Kosong	21
Tabel 6. Rancangan Pengujian Daftar – Username tidak sesuai format	22
Tabel 7. Rancangan Pengujian Daftar – Nama Tidak Sesuai Format.....	24
Tabel 8. Rancangan Pengujian Daftar – Email sudah terdaftar	26
Tabel 9. Rancangan Pengujian Daftar – Email tidak sesuai format.....	28
Tabel 10. Rancangan Pengujian Daftar – Kata Sandi < 8 karakter.....	30
Tabel 11. Rancangan Pengujian Daftar – Kata Sandi Tidak Sesuai Format.....	32
Tabel 12. Rancangan Pengujian Daftar – Konfirmasi Kata Sandi Salah.....	34
Tabel 13. Rancangan Pengujian Daftar – Nomor Ponsel Tidak Valid	36
Tabel 14. Rancangan Pengujian Login – Alur Utama Positif.....	38
Tabel 15. Rancangan Pengujian Login – Alur Utama Positif Menggunakan Username	39
Tabel 16. Rancangan Pengujian Login – Salah Inputan Email.....	40
Tabel 17. Rancangan Pengujian Login – Salah Username	41
Tabel 18. Rancangan Pengujian Login – Salah Password	42
Tabel 19. Rancangan Pengujian Login – Username Kosong.....	43
Tabel 20. Rancangan Pengujian Login – Password Kosong.....	44
Tabel 21. Rancangan Pengujian Logout – Alur Utama Positif.....	45
Tabel 22 .Rancangan Pengujian Lupa Kata Sandi – Alur Utama Positif.....	46
Tabel 23. Rancangan Pengujian Lupa Kata Sandi – Email Tidak Valid	48
Tabel 24. Rancangan Pengujian Lupa Kata Sandi – Email Tidak Sesuai Format .	50
Tabel 25. Rancangan Pengujian Lupa Kata Sandi – Email Kosong	51
Tabel 26. Rancangan Pengujian Lupa Kata Sandi - OTP Salah	52
Tabel 27. Rancangan Pengujian Lupa Kata Sandi – OTP Di Blok.....	54
Tabel 28. Rancangan Pengujian Lupa Kata Sandi – Field Kosong	56
Tabel 29. Rancangan Pengujian Lupa Kata Sandi – Konfirmasi Kata Sandi Baru	

Tidak Sesuai	58
Tabel 30. Rancangan Pengujian Lupa Kata Sandi – Kata Sandi Baru Tidak Sesuai Format.....	60
Tabel 31. Rancangan Pengujian Ubah Kata Sandi – Alur Utama Positif	62
Tabel 32. Rancangan Pengujian Ubah Kata Sandi – Kata Sandi Lama Salah	64
Tabel 33. Rancangan Pengujian Ubah kata Sandi – Kata Sandi Lama Kosong	65
Tabel 34. Rancangan Pengujian Ubah kata Sandi – Kata Sandi Baru < 8 Karakter	67
Tabel 35. Rancangan Pengujian Ubah kata Sandi – Kata Sandi Baru Kosong.....	69
Tabel 36. Rancangan Pengujian Ubah kata Sandi – Kata Sandi Baru Tidak Sesuai Format.....	70
Tabel 37. Rancangan Pengujian Ubah kata Sandi – Konfirmasi Kata Sandi Baru Kosong.....	72
Tabel 38. Rancangan Pengujian Ubah kata Sandi – Konfirmasi Kata Sandi Baru Tidak Sesuai	74
Tabel 39. Rancangan Pengujian Hubungi Kami – Alur Utama Positif.....	76
Tabel 40. Rancangan Pengujian Hubungi Kami – Nomor Ponsel Kosong.....	77
Tabel 41. Rancangan Pengujian Hubungi Kami – Pesan Kosong	78
Tabel 42. Rancangan Pengujian Hubungi Kami – Nomor Ponsel Campur	79
Tabel 43. Rancangan Pengujian Hubungi Kami – Nomor Ponsel Tidak Sesuai Format.....	81
Tabel 44. Rancangan Pengujian Hubungi Kami – Pesan < 10 Karakter	82
Tabel 45. Rancangan Pengujian Edit Profil – Alur Utama Positif.....	83
Tabel 46. Rancangan Pengujian Edit Profil – Nomor Ponsel Tidak Diawali 08 ...	85
Tabel 47. Rancangan Pengujian Edit Profil – Nomor Ponsel Tidak Valid	86
Tabel 48. Rancangan Pengujian Edit Profil – Nomor Ponsel Campur Karakter ...	87
Tabel 49. Rancangan Pengujian Edit Profil – Alamat Tidak Valid	89
Tabel 50. Rancangan Pengujian Edit Profil – Umur < 17 Tahun	90
Tabel 51. Rancangan Pengujian Edit Profil – Pekerjaan Tidak Valid	92
Tabel 52. Rancangan Pengujian Tambah Akun Bank – Alur Utama Positif	94
Tabel 53. Rancangan Pengujian Tambah Akun Bank – Alur Utama Positif Tambah	

Kedua.....	96
Tabel 54. Rancangan Pengujian Tambah Akun Bank – Nama Bank Kosong	98
Tabel 55. Rancangan Pengujian Tambah Akun Bank – Cabang Kosong.....	100
Tabel 56. Rancangan Pengujian Tambah Akun Bank – Nama Pemilik Rekening Kosong.....	101
Tabel 57. Rancangan Pengujian Tambah Akun Bank – Nama Pemilik < 3 Karakter	103
Tabel 58. Rancangan Pengujian Tambah Akun Bank – nama Pemilik Rekening Campur Angka.....	105
Tabel 59. Rancangan Pengujian Tambah Akun Bank - Nomor Rekening Kosong	107
Tabel 60. Rancangan Pengujian Tambah Akun Bank – Nomor Rekening Pendek	109
Tabel 61. Rancangan Pengujian Tambah Akun Bank – Nomor Rekening Campur Huruf.....	111
Tabel 62. Rancangan Pengujian Tambah Akun Bank – Kata Sandi Akun Salah	113
Tabel 63. Rancangan Pengujian Verifikasi Nomor Ponsel – Alur Utama Positif	116
Tabel 64. Rancangan Pengujian Verifikasi Nomor Ponsel – OTP Kosong.....	118
Tabel 65. Rancangan Pengujian Verifikasi Nomor Ponsel – OTP Salah	119
Tabel 66. Rancangan Pengujian Verifikasi Nomor Ponsel – OTP Campur Huruf	121
Tabel 67. Rancangan Pengujian Verifikasi Nomor Ponsel – OTP Salah 3 Kali..	122
Tabel 68. Rancangan Pengujian Kalkulator Kredit – Alur Utama Positif.....	125
Tabel 69. Rancangan Pengujian Kalkulator Kredit – Alur Utama Positif Kedua	127
Tabel 70. Rancangan Kalkulator Kredit – Harga Mobil Campur Huruf.....	128
Tabel 71. Rancangan Pengujian Kalkulator Kredit – Uang Muka Campur Huruf	130
Tabel 72. Rancangan Pengujian Kalkulator Kredit – Uang Muka > Harga Mobil	131
Tabel 73. Rancangan Pengujian Kalkulator Kredit – Ulang Kalkulasi	133
Tabel 74. Rancangan Pengujian Kalkulator Budget – Alur Utama Positif.....	134

Tabel 75. Rancangan Pengujian Kalkulator Budget – Uang Muka Campur Huruf	136
Tabel 76. Rancangan Pengujian Kalkulator Budget – Cicilan Bulanan Campur Huruf.....	137
Tabel 77. Hasil Eksekusi Daftar – Alur Utama Positif	139
Tabel 78. Hasil Eksekusi Daftar – Inputan Kosong	141
Tabel 79. Hasil Eksekusi Daftar – Username Tidak Sesuai Format	142
Tabel 80. Hasil Eksekusi Daftar – Nama Tidak Sesuai Format.....	144
Tabel 81. Hasil Eksekusi Daftar – Email Sudah Terdaftar	146
Tabel 82. Hasil Eksekusi Daftar – Email Tidak Sesuai Format.....	148
Tabel 83. Kata Sandi Kurang < 8 Karakter.....	150
Tabel 84. Hasil Eksekusi Daftar – Kata Sandi Tidak Sesuai Format.....	152
Tabel 85. Hasil Eksekusi Daftar – Konfirmasi Kata Sandi Tidak Sesuai	154
Tabel 86. Hasil Eksekusi Daftar – Nomor Ponsel Tidak Valid	156
Tabel 87. Hasil Eksekusi Login – Menggunakan Email.....	158
Tabel 88. Hasil Eksekusi Login – Menggunakan Username	159
Tabel 89. Hasil Eksekusi Login – Salah Inputan Email	160
Tabel 90. Hasil Eksekusi Login – Salah Inputan Username.....	161
Tabel 91. Hasil Eksekusi Login – Salah Inputan Kata Sandi	162
Tabel 92. Hasil Eksekusi Login – Username Kosong.....	163
Tabel 93. Hasil Eksekusi Login – Inputan Kata Sandi Kosong.....	164
Tabel 94. Hasil Eksekusi Logout – Alur Utama Positif.....	165
Tabel 95. Hasil Eksekusi Lupa Kata Sandi – Alur Utama Positif	166
Tabel 96. Hasil Eksekusi Lupa Kata Sandi – Email Akun Tidak Valid	168
Tabel 97. Hasil Eksekusi Lupa Kata Sandi – Email Akun Tidak Sesuai Format	169
Tabel 98. Hasil Eksekusi Lupa Kata Sandi – Email Kosong.....	170
Tabel 99. Hasil Eksekusi Lupa Kata Sandi – OTP Salah	171
Tabel 100. Hasil Eksekusi Lupa Kata Sandi – OTP Salah 3 Kali.....	173
Tabel 101. Hasil Eksekusi Lupa Kata Sandi – Kata Sandi Baru dan Konfirmasi Kata Sandi Kosong.....	175
Tabel 102. Hasil Eksekusi Lupa Kata Sandi – Konfirmasi Kata Sandi Baru Tidak	

Sesuai.....	177
Tabel 103. Hasil Eksekusi Lupa Kata Sandi – Kata Sandi Baru Tidak Sesuai Format	179
Tabel 104. Hasil Eksekusi Ubah Kata Sandi – Alur Utama Positif.....	181
Tabel 105. Hasil Eksekusi Ubah Kata Sandi – Kata Sandi Lama Tidak Sesuai ..	182
Tabel 106. Hasil Eksekusi Ubah Kata Sandi – Kata Sandi Lama Kosong	184
Tabel 107. Kata Sandi Baru < 8 Karakter	185
Tabel 108. Hasil Eksekusi Ubah Kata Sandi – Kata Sandi Baru Kosong.....	187
Tabel 109. Hasil Eksekusi Ubah Kata Sandi – Kata Sandi Baru Tidak Sesuai Format	188
Tabel 110. Hasil Eksekusi Ubah Kata Sandi – Konfirmasi Kata Sandi Baru Kosong	190
Tabel 111. Hasil Eksekusi Ubah Kata Sandi – Konfirmasi Kata Sandi Baru Tidak Sesuai.....	191
Tabel 112. Hasil Eksekusi Hubungi Kami – Alur Utama Positif	193
Tabel 113. Hasil Eksekusi Hubungi Kami – Nomor Ponsel Kosong	194
Tabel 114. Hasil Eksekusi Hubungi Kami – Pesan Kosong	195
Tabel 115. Hasil Eksekusi Hubungi Kami – Nomor Ponsel Campur Karakter ...	196
Tabel 116. Hasil Eksekusi Hubungi Kami – Nomor Ponsel Tidak Sesuai Format	198
Tabel 117. Hasil Eksekusi Hubungi Kami – Pesan < 10 Karakter	199
Tabel 118. Hasil Eksekusi Edit Profil – Alur Utama Positif	200
Tabel 119. Hasil Eksekusi Edit Profil – Nomor Ponsel Tidak Diawali 08	202
Tabel 120. Hasil Eksekusi Edit Profil – Nomor Ponsel Tidak Valid.....	203
Tabel 121. Hasil Eksekusi Edit Profil – Nomor Ponsel Campur Karakter	204
Tabel 122. Hasil Eksekusi Edit Profil – Alamat Tidak Valid	205
Tabel 123. Hasil Eksekusi Edit Profil – Umur < 17 Tahun	207
Tabel 124. Hasil Eksekusi Edit Profil – Pekerjaan Tidak Valid	209
Tabel 125. Hasil Eksekusi Tambah Akun Bank – Alur Utama Positif.....	211
Tabel 126. Hasil Eksekusi Tambah Akun Bank – Alur Utama Positif Kedua	213
Tabel 127. Hasil Eksekusi Tambah Akun Bank – Nama Bank Kosong.....	215

Tabel 128. Hasil Eksekusi Tambah Akun Bank – Cabang Kosong.....	216
Tabel 129. Hasil Eksekusi Tambah Akun Bank – Nama Pemilik Rekening Kosong	217
Tabel 130. Hasil Eksekusi Tambah Akun Bank – Nama Pemilik Rekening < 3 Karakter.....	219
Tabel 131. Hasil Eksekusi Tambah Akun Bank – Nama Pemilik Rekening Campur Angka.....	221
Tabel 132. Hasil Eksekusi Tambah Akun Bank – Nomor Rekening Kosong.....	223
Tabel 133. Hasil Eksekusi Tambah Akun Bank – Nomor Rekening Pendek.....	224
Tabel 134. Hasil Eksekusi Tambah Akun Bank – Nomor Rekening Campur Huruf	226
Tabel 135. Hasil Eksekusi Tambah Akun Bank – Kata Sandi Akun Salah.....	228
Tabel 136. Hasil Eksekusi Verifikasi Nomor Ponsel – Alur Utama Positif.....	230
Tabel 137. Hasil Eksekusi Verifikasi Nomor Ponsel – OTP Kosong.....	232
Tabel 138. Hasil Eksekusi Verifikasi Nomor Ponsel – OTP Salah.....	233
Tabel 139. Hasil Eksekusi Verifikasi Nomor Ponsel – OTP Campur Huruf.....	235
Tabel 140. Hasil Eksekusi Verifikasi Nomor Ponsel – OTP Salah 3 Kali.....	237
Tabel 141. Hasil Eksekusi Kalkulator Kredit – Alur Utama Positif.....	239
Tabel 142. Hasil Eksekusi Kalkulator Kredit – Alur Utama Positif Kedua.....	241
Tabel 143. Hasil Eksekusi Kalkulator Kredit – Harga Mobil Campur Huruf.....	242
Tabel 144. Hasil Eksekusi Kalkulator Kredit – Uang Muka Campur Huruf.....	244
Tabel 145. Hasil Eksekusi Kalkulator Kredit – Uang Muka > Harga Mobil.....	246
Tabel 146. Hasil Eksekusi Kalkulator Kredit – Ulang Kalkulasi.....	247
Tabel 147. Hasil Eksekusi Kalkulator Budget – Alur Utama Positif.....	249
Tabel 148. Hasil Eksekusi Kalkulator Budget – Uang Muka Campur Huruf.....	250
Tabel 149. Hasil Eksekusi Kalkulator Budget – Cicilan Bulanan Campur Huruf	252
Tabel 150. Perbandingan Pelaksanaan Pengujian.....	278
Tabel 151. Perbandingan Waktu Pelaksanaan Pengujian.....	279
Tabel 152. Perbandingan Start Application Hubungi Kami.....	281
Tabel 153. Perbandingan Start Application Kalkulator Budget.....	282

INTISARI

ANALISIS PERBANDINGAN BLACK-BOX AUTOMATED TESTING DAN MANUAL TESTING PADA APLIKASI ACCMART

Hartono Rusli

150708608

Peningkatan penggunaan *smartphone* pada beberapa tahun terakhir menyebabkan meningkatnya kebutuhan akan aplikasi *mobile*. Hal ini pun mempengaruhi perusahaan-perusahaan besar termasuk PT. Astra Credit Companies. Untuk mengikuti perkembangan aplikasi yang sedang meningkat, maka PT. Astra Credit Companies pun meluncurkan aplikasi ACCMart. Karena itu diperlukan pengujian pada aplikasi ACCMart untuk memastikan aplikasi berjalan dengan lancar. Selain itu perlu juga menggunakan metode dan cara pengujian yang tepat serta metode pengujian yang paling tepat untuk melakukan pengujian.

Pada penelitian ini pengujian terhadap aplikasi ACCMart dilakukan secara otomatis dan manual. Kemudian tipe pengujian yang dilakukan adalah *black-box testing* terhadap 11 dari 25 fungsi-fungsi yang ada. Pada penelitian ini, akan diteliti cara-cara melakukan pengujian terhadap aplikasi ACCMart. Selain itu di dalam penelitian juga akan melakukan perbandingan antara keduanya. Perbandingan dilakukan dengan menggunakan selisih waktu pengujian sebagai parameter. Pengujian manual akan dilakukan langsung dan otomatis dengan Katalon Studio.

Hasil yang telah didapatkan dari penelitian adalah cara pengujian manual dilakukan secara langsung oleh penguji. Semua penilaian dilakukan oleh penguji secara langsung. Sementara pengujian otomatis dilakukan di dalam Katalon Studio setelah penguji merancang *test case*, *test suite*, dll. Setelah pengujian dilakukan dan dibandingkan, selisih waktu yang diperlukan untuk pengujian manual adalah 61,47%. Hal ini berarti pengujian manual lebih cepat 61,47% dibandingkan pengujian otomatis. Selain itu, dari 11 fungsi yang diuji, ditemukan total 10 *bug* dari 4 fungsi di antaranya

Kata Kunci: Katalon Studio, *Black-Box*, ACCMart, pengujian otomatis, pengujian manual

Dosen Pembimbing I : Martinus Maslim, S.T., M.T.

Dosen Pembimbing II : Stephanie Pamela Adithama, S.T., M.T.

Jadwal Sidang Tugas Akhir : 3 Juni 2020

BAB I. PENDAHULUAN

1.1. Latar Belakang

Perangkat *mobile* sangatlah umum digunakan pada zaman sekarang karena mudah dibawa, mudah digunakan dan mudah diakses. Perangkat *mobile* sendiri telah berkontribusi miliaran uang terhadap ekonomi dunia. Di mana ekonomi tersebut tercipta dari banyaknya aplikasi *mobile*, pengembang, dan perusahaan. Berdasarkan dari statistik dari statista.com, sejak tahun 2019, ada lebih dari 2 hingga 2.5 juta aplikasi yang tersedia bagi para pengguna Android. Diikuti Apple's *app store* di belakangnya dengan hampir 2 juta aplikasi *mobile*. Pengembangan aplikasi *mobile* itu sendiri didorong oleh tumbuhnya kebutuhan yang mengharuskan orang-orang untuk dapat bekerja dan mengakses segala sesuatu dari mana saja dan kapan saja. Kebutuhan ini menimbulkan perlunya fungsionalitas yang jauh lebih kompleks dan pengembangan yang lebih cepat pada sisi perangkat lunak maupun perangkat keras dibanding sebelumnya. Sebuah aplikasi akan jauh lebih populer jika aplikasi tersebut bebas dari *bugs*, memiliki tingkat fungsionalitas yang tinggi, dapat diandalkan dan mudah digunakan. Jika suatu aplikasi menyimpang dari beberapa aspek yang disebutkan sebelumnya, maka kemungkinan aplikasi tersebut menjadi terbuang sangatlah tinggi.

Pengujian atau *testing* merupakan bagian penting dari pengembangan aplikasi *mobile*. *Testing* atau pengujian sendiri pada umumnya dibagi ke dalam 2 metode. Metode yang pertama yaitu pengujian manual. Pengujian manual dilakukan secara langsung oleh *tester* atau penguji yang berperan sebagai pengguna yang berinteraksi dengan sistem dan mencoba berbagai fungsionalitas yang ada di dalam aplikasi. Selain itu metode ini juga digunakan untuk melakukan tugas-tugas yang membutuhkan logika manusia. pengujian manual juga memerlukan banyak sekali sumber daya manusia dan terkadang dapat menjadi tidak konsisten karena kesalahan manusia [1]. Metode yang kedua adalah metode yang diketahui sebagai *mobile test automation*. *Automated testing* dilakukan dengan menggunakan perangkat lunak untuk pengujian dengan menciptakan *test case* untuk masing-

masing alur dan fungsionalitas dari aplikasi. Hal ini membantu untuk mengurangi waktu yang diperlukan untuk melakukan tes yang sebelumnya dilakukan secara manual oleh manusia dan mengurangi kurangnya konsistensi karena tidak perlu ada manusia yang terlibat dalam tes selain melakukan desain *test case* [2].

Sebuah aplikasi tidak mungkin terlepas dari *error* atau *bugs*. Hal ini dikarenakan saat pengembangan jarang sekali tidak terjadi kesalahan, baik dari tahap pengumpulan kebutuhan hingga pengembangan. Waktu yang diperlukan untuk melakukan pengujian untuk mencari *bugs* pun sangat banyak. Bahkan berdasarkan [3] lebih dari 40% waktu pengembangan dilakukan untuk pengujian. Maka dari itu sangatlah penting untuk melakukan pengujian terhadap aplikasi yang dikembangkan.

Pengujian manual dan otomatis akan dilakukan pada aplikasi ACCMart. Aplikasi ACCMart merupakan aplikasi yang dibuat oleh perusahaan Astra Credit Companies untuk transaksi jual beli mobil dan lelang. Pengujian perlu dilakukan untuk mencari *bugs* pada aplikasi ini karena masih dalam tahap pengembangan. Metode yang dilakukan untuk pengujian adalah metode manual dan otomatis secara black-box. Alasan penggunaan metode black-box adalah karena pengembangan aplikasi sudah mencapai tahap akhir dan memasuki fase UAC (*User Acceptance Test*) sehingga perlu pengujian dengan peran penguji sebagai pengguna. Pengujian manual dan otomatis diperlukan untuk mencari tahu metode mana yang paling efisien dan cocok digunakan. Untuk pengujian otomatis diperlukan alat untuk merancang pengujian tersebut. Katalon Studio merupakan alat yang tepat karena alat ini merupakan alat yang relatif baru dan mudah untuk digunakan para penguji aplikasi.

1.2. Rumusan Masalah

Adapun rumusan masalah dalam penelitian ini adalah:

1. Bagaimana cara melakukan pengujian manual oleh user dan otomatis dengan menggunakan Katalon Studio pada aplikasi ACCMart?
2. Bagaimanakah perbandingan antara pengujian otomatis dengan menggunakan Katalon Studio dan manual pada aplikasi ACCMart?

1.3. Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah:

1. Aplikasi yang akan diuji adalah aplikasi ACCMart.
2. Fungsi – fungsi yang akan diuji telah ditentukan sebelumnya dan hanya 11 dari 25 fungsi yang ada.
3. Alat pengujian otomatis yang akan digunakan adalah Katalon Studio.
4. Metode yang digunakan adalah metode black-box

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Melakukan pengujian manual dan otomatis pada aplikasi ACCMart
2. Menganalisis perbandingan antara pengujian otomatis menggunakan Katalon Studio dan manual oleh user

1.5. Metode Penelitian

Metode yang akan digunakan dalam penelitian ini adalah:

1. Analisis *functional flow* pada fungsi yang akan diuji

Langkah pertama dalam penelitian ini adalah melakukan sebuah analisis terhadap alur fungsi yang akan diuji. Hal ini dilakukan dengan melihat dokumen yang disediakan oleh pembuat aplikasi atau dengan uji manual.

2. Perancangan Skenario Pengujian

Dalam langkah ini, penulis akan membuat skenario pengujian. Skenario pengujian merupakan pengelompokan pengujian berdasarkan fungsi – fungsi yang akan diuji dalam penelitian.

3. Perancangan *Test Case*

Langkah ini merupakan langkah pembuatan *test case* yang menjadi bagian dari *test scenario*. *Test Case* akan dirancang dan dibagi menjadi dua bagian, *Positive testing* dan *Negative testing*. *Positive testing* merupakan pengujian yang menggunakan parameter yang benar dan

sesuai dengan tujuan dari fungsi yang ada. Tujuannya adalah untuk menguji alur utama fungsi. *Negative testing* merupakan pengujian yang menggunakan parameter yang tidak benar dan tidak sesuai dengan tujuan dari fungsi yang ada. Tujuan dari *negative testing* adalah untuk menguji *exception handling* dari aplikasi.

4. Pelaksanaan pengujian manual

Dalam langkah ini, penulis akan melakukan pengujian manual sesuai dengan skenario pengujian dan *test case* yang telah dirancang sebelumnya.

5. Pelaksanaan pengujian otomatis

Dalam langkah ini, penulis akan melakukan pengujian otomatis dengan menggunakan Katalon Studio. Pengujian akan dilakukan sesuai dengan skenario pengujian dan *test case* yang telah dirancang.

6. Membandingkan pengujian manual dan otomatis

Langkah ini akan membandingkan hasil antara pengujian manual dan otomatis. Perbandingan yang akan dilakukan antara lainnya adalah seperti metode mana yang dapat digunakan untuk pengujian fungsi tertentu, metode mana yang lebih cepat, dll.

7. Penulisan laporan akhir

Langkah ini merupakan langkah terakhir dalam penelitian ini. Dalam langkah ini, penulis akan menyusun laporan sesuai dengan hasil eksperimen yang telah dilakukan dalam penelitian.

1.6. Sistematika Penulisan

Laporan ini disusun secara sistematis dengan urutan sebagai berikut:

BAB 1 PENDAHULUAN

Bab ini tersusun dari latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian, dan sistematika penulisan.

BAB 2 TINJAUAN PUSTAKA

Bab ini tersusun dari penjelasan dari penelitian – penelitian yang telah dilakukan sebelumnya untuk pemecahan masalah dalam laporan ini.

BAB 3 LANDASAN TEORI

Bab ini tersusun dari dasar teori yang digunakan sebagai dasar dan acuan dalam pemecahan masalah.

BAB 4 ANALISIS DAN PERANCANGAN EKSPERIMEN

Bab ini tersusun dari analisis masalah yang dihadapi dan tahap perancangan eksperimen yang akan dilakukan untuk memecahkan masalah.

BAB 5 HASIL EKSPERIMEN DAN PEMBAHASAN

Bab ini tersusun dari hasil dari eksperimen yang telah dilakukan dan pembahasan mengenai hasil eksperimen.

BAB 6 PENUTUP

Bab ini berisi kesimpulan mengenai eksperimen yang telah dilakukan.

BAB II. TINJAUAN PUSTAKA

Pada bab ini akan dijelaskan hal-hal yang berkaitan dengan penelitian yang akan dilakukan serta perbandingan dengan penelitian-penelitian yang telah dilakukan sebelumnya.

Dalam jurnal yang berjudul *Automated testing of Mobile Applications Using Scripting Technique: A Study on Appium* oleh Shiwangi, et al, membahas tentang metode pengujian yang menggunakan *scripting*. Penelitian yang dilakukan juga melihat aspek antara pengujian yang dilakukan secara manual dan juga pengujian yang dilakukan secara otomatis. Berdasarkan Ilene Burnstein pengujian perangkat lunak merupakan sekelompok prosedur yang digunakan untuk menguji aspek – aspek dari perangkat lunak [4]. Pengujian perangkat lunak pada umumnya dibagi menjadi 2 kategori utama: manual dan *automated testing*. Dalam pengujian manual tidak ada pemrograman yang bisa dilakukan untuk menciptakan testing rumit yang dapat mencatat kesalahan yang terjadi dan memberi tahu informasi yang hilang. Sementara di *automated testing*, *tester* dapat menciptakan pengujian yang lebih rumit yang dapat memberi tahu informasi yang hilang dan kesalahan – kesalahan yang terjadi. Karhu menyimpulkan perbedaan dari kedua kategori ini dengan mengatakan bahwa *automated testing* lebih baik digunakan untuk mencegah munculnya kesalahan baru di modul yang sudah diuji sebelumnya, sementara pengujian manual lebih baik digunakan untuk menemukan kesalahan yang tidak terduga dan baru [5].

Dalam jurnalnya yang berjudul “*Mobile Application Testing Matrix and Challenges*” oleh Amen, et al, membahas tentang metode pengujian dan tantangan dalam penggunaannya. Dalam jurnal ini, kedua metode manual dan juga *automated* akan dibahas. Dalam risetnya, *automated testing* dapat mengurangi kesalahan akibat manusia yang muncul, efisiensi dalam menemukan *bugs*, dengan waktu yang lebih singkat. Sementara itu, metode manual sangat menghabiskan waktu

dibandingkan *automated testing*, dan seringkali memiliki batasan dalam pengujian karena keterbatasan antarmuka di perangkat. *Manual testing* juga lebih menghabiskan waktu dibandingkan dengan *automated testing* dan dibatasi oleh terbatasnya antar-muka pengguna di perangkat [6]. Selain itu, Pengujian perangkat lunak juga lebih rumit dibandingkan *software* atau aplikasi web karena perbedaan pengembangan seperti; OS, perangkat dan resolusi layar [7]. Perangkat pengujian yang digunakan dalam jurnal ini adalah MonkeyTalk yang dirilis oleh Gorilla Logic. Hasil dari penelitian yang dilakukan adalah dari beberapa alat yang digunakan, salah satunya Monkey Talk merupakan beberapa alat yang dapat digunakan untuk melakukan pengujian fungsionalitas dan juga GUI. *Automated testing* merupakan solusi untuk meningkatkan efisiensi pengujian. Selain itu metode ini juga sangat penting untuk meningkatkan pengujian fungsionalitas sebagai alat untuk menangani pengujian di beberapa perangkat yang memiliki tipe yang berbeda. Keuntungan dari menggunakan *automated testing* adalah lebih banyak pengujian yang dapat dilakukan dalam waktu yang lebih singkat dan dapat mengurangi waktu dan biaya [8].

Dalam jurnalnya yang berjudul “*Automated Tests for Mobile Games: an Experience Report*” oleh Lovreto, et al, membahas tentang penerapan *automated testing* dalam pengujian game. Pengujian game pada umumnya menggunakan metode manual dan karena itu sangat bergantung terhadap *tester* manusia Hal ini berakibat terhadap meningkatnya kesalahan dan biaya yang dibutuhkan Di dalam jurnal ini ada 16 game populer yang dipilih untuk diuji. Dalam pengembangan sebuah game ada banyak teknik yang digunakan untuk menangani pengembangan yang rumit. Pengujian perangkat lunak merupakan salah satu bagian penting untuk menjamin standar kualitas terpenuhi dan mengurangi usaha untuk perawatan dan pengembangan secara keseluruhan. Myers, et al, menyimpulkan bahwa pengujian perangkat lunak merupakan “proses mengeksekusi program dengan tujuan menemukan kesalahan [9]. Hasil yang didapatkan dari penelitian yang dilakukan di jurnal ini adalah bahwa *automated testing* sangat mempengaruhi efisiensi dan menghemat waktu [10].

Dalam jurnal yang berjudul "*Analysis of Efficiency of Automated Software Testing Methods: Direction of Research*" oleh Valliammai, et al, membahas tentang efisiensi metode pengujian. Berdasarkan jurnal ini, teknik pengujian yang paling efektif memperlihatkan angka maksimal error dan menginspirasi tingkat maksimum dari keyakinan akan kebenaran program [11]. Zaman sekarang, pengujian dalam pengembangan perangkat lunak memiliki peran yang sangat penting. Hal ini dikarenakan setiap tahunnya ada jumlah yang luar biasa banyak yang dipasarkan dengan prediksi 76,9 milyar *download* secara global di tahun 2014 [12]. Pengujian menjamin perangkat lunak memenuhi keinginan dan spesifikasi dari pengguna [13]. Dapat dilihat dari jumlah uang dan juga biaya yang diinvestasikan ke dalam pengujian perangkat lunak. Survey yang telah dilakukan di dalam jurnal ini adalah berdasarkan dari standar fungsional dan praktikal yang telah ditunjukkan atau diilustrasikan oleh perangkat pengujian bahwa pengguna dapat menggunakan perangkat pengujian dengan mudah dalam suatu aplikasi. Kesimpulan yang didapat dari penelitian yang telah dilakukan dalam jurnal ini adalah 3 karakteristik untuk melakukan perbandingan metode pengujian. Yaitu; yang pertama cakupan kode yang luas, semakin luas kode yang dicakup maka semakin baik metode yang digunakan tersebut, kedua kriteria yang dicakup, semakin banyak kriteria pengujian yang dapat dilakukan, maka metode tersebut akan semakin baik. Terakhir, merupakan otomatisasi dan laporan. Jika metode tersebut sudah otomatis dan memiliki fungsi untuk membuat sebuah laporan maka metode tersebut akan semakin baik.

Dalam jurnal yang berjudul "*Review of Mobile Applications Testing with Automated Techniques*" oleh Kaur, membahas teknik otomatisasi yang digunakan untuk melakukan pengujian terhadap aplikasi. Dalam perkembangan pesat ini, sangatlah penting untuk fokus dalam pengujian fungsional, pengujian keamanan, pengujian performa, pengujian penggunaan, pengujian regresi dan pengujian kompatibilitas [14]. Pengujian juga merupakan salah satu faktor yang sangat penting dalam meningkatkan kualitas aplikasi. Aplikasi yang diciptakan juga semakin

kompleks, sesuai dengan kinerja perangkat keras yang digunakan. Selain itu, dalam jurnal ini juga disebutkan beberapa teknik yang tersedia untuk melakukan pengujian terhadap aplikasi. Beberapa teknik yang dapat digunakan adalah *keyword-driven testing* [15] yang menggunakan Robotium yang dikombinasikan dengan kerangka pengujian *keyword-driven* yang memisahkan logika pengujian, skrip pengujian, dan data pengujian. *Test driven mobile applications development* [16] memberikan teknik dimana ketika mengembangkan aplikasi, pengujian otomatis dijalankan bersamaan dengan pengembangan. Selain itu teknik *GUI Crawling-Based technique* untuk aplikasi android oleh Amalfitano [17] merupakan teknik yang mengeksekusi pengujian GUI dalam aplikasi android. Tujuan utamanya adalah untuk menemukan kesalahan ketika aplikasi digunakan. Terakhir, adalah teknik yang tidak memerlukan koneksi antara perangkat *mobile* dan komputer. Dalam teknik ini, pengujian dilakukan di perangkat itu sendiri karena sebagian pengujian tidak dapat disimulasikan [18].

Pada tabel 1, dapat dilihat perbandingan antara penelitian yang dilakukan dengan penelitian – penelitian yang telah dilakukan sebelumnya.

Tabel 1. Perbandingan Penelitian Dengan Penelitian Sebelumnya

Sumber	Shiwangi, et al [1]	Amen, et al., [8]	Lovreto, et al., [10]	Valliammai, et al, [11]	Rusli (2020)*
Judul	<i>Automated Testing of Mobile Applications using Scripting Technique: A Study on Appium</i>	<i>Mobile Testing Matrix and Challenges</i>	<i>Automated Tests for Mobile Games An Experience Report</i>	<i>Testing of Mobile Applications. A Review Of Practices</i>	Perbandingan Black-Box Automated Testing dan Manual Testing Pada Aplikasi ACCMart
Script	Ruby, dll.	MonkeyTalk Script / Javascript	Python	-	Groovy
Tools	Appium	MonkeyTalk	Appium	-	Katalon Studio
Metode	<i>Automated Testing</i>	<i>Automated Testing dan Manual Testing</i>	<i>Automated Testing</i>	<i>Analisis Manual Testing</i>	<i>Automated Testing dan Manual Testing</i>

***)Penelitian yang dilakukan**

BAB III. LANDASAN TEORI

3.1. ACCMart

ACCMart merupakan aplikasi yang diciptakan sebagai sebuah fasilitas untuk kebutuhan – kebutuhan pembeli dan penjual terhadap kendaraan bermotor. Selain itu kendaraan yang dijual juga hanya dapat diakses atau dibeli oleh pengguna yang telah terdaftar. Aplikasi ini berfungsi sebagai sebuah pasar untuk memperjualkan kendaraan bermotor. Dalam aplikasi ini juga memiliki fungsi lelang untuk melelang mobil bekas yang ditarik dari pelanggan. Pada ACCMart terdapat banyak fungsi – fungsi yang dapat digunakan pengguna, salah satunya adalah fungsi kalkulator yang dapat digunakan oleh pengguna untuk melakukan kalkulasi cicilan bulanan terhadap kendaraan yang diinginkan, atau mencari kendaraan sesuai dengan *budget* yang dimiliki. Selain itu, salah satu fungsi utama ACCMart adalah untuk melakukan penjualan kendaraan bermotor. Pengguna dapat mengakses kendaraan – kendaraan bermotor yang sedang dijual dan membelinya melalui aplikasi ini. ACCMart juga memiliki fungsi jual yang dapat digunakan oleh user untuk menjual kendaraan yang dimilikinya langsung melalui aplikasi ini, namun fitur ini masih belum tersedia hingga sekarang. ACCMart sendiri memiliki secara keseluruhan 25 fungsi. Fungsi fungsi tersebut antara lain adalah Daftar, Login, Logout, Ubah Kata Sandi, Lupa Kata Sandi, Edit Profil, Hubungi Kami, Tambah Akun Bank, Kalkulator Kredit, Kalkulator Budget, Verifikasi Nomor Ponsel, Cari Mobil, Add Wishlist, Bandingkan Mobil, Kirim Pesan, Upload Dokumen, Bayar Commitment Fee, Lihat Promo, Lihat Semua Produk hot deals, Beli Sekarang, Keranjang, Edit Akun Bank, Riwayat Pembelian, Upload Profile Picture, dan Jual.

3.2. Software Bug

Merupakan sebuah kesalahan, kegagalan atau cacat dalam suatu program komputer atau sistem yang menyebabkan hasil yang salah atau tidak terduga, atau menyebabkan perilaku yang tidak diinginkan. Proses mencari dan memperbaiki *bugs* seringkali disebut dengan *debugging* dan sering menggunakan suatu alat bantu

untuk membantu. Sebagian besar *bugs* muncul dari kesalahan yang tercipta dalam kode atau desain suatu perangkat lunak, atau dalam komponen dan sistem operasi yang digunakan oleh suatu program. *Bugs* juga dapat memicu error yang dapat memiliki efek yang berkelanjutan. Selain itu, *bugs* juga memiliki efek yang halus atau berat seperti menyebabkan komputer untuk berhenti berjalan sepenuhnya. Untuk menangani *bugs* diperlukan manajemen yang meliputi dokumentasi, kategorisasi, memperbaiki, dll. Pada umumnya *bugs* dilacak dengan menggunakan sebuah sistem pelacak atau sistem pencari kesalahan [19].

3.3. Software Testing

Merupakan sebuah tahap pengembangan yang digunakan untuk menyediakan informasi mengenai kualitas produk perangkat lunak yang telah dibuat [20]. Pengujian perangkat lunak juga dapat menyediakan pandangan yang objektif dan independen agar pihak bisnis dapat memahami resiko dari implementasi perangkat lunak. Teknik yang digunakan adalah melakukan eksekusi dari program dengan tujuan untuk menemukan kesalahan atau cacat di dalam produk. Selain itu pengujian perangkat lunak juga mencakup eksekusi dari komponen perangkat lunak atau komponen sistem untuk melakukan evaluasi dari satu atau lebih property. Pengujian perangkat lunak dapat menyediakan informasi mengenai semua kesalahan atau cacat yang ada di dalam produk. Tidak semua kesalahan perangkat lunak disebabkan oleh kesalahan pemrograman, salah satu penyebabnya adalah kesalahan pada saat pengumpulan kebutuhan, contohnya seperti kebutuhan yang terlupakan yang pada akhirnya menjadi sebuah kesalahan [21]. Pengujian juga dapat dilakukan segera setelah perangkat lunak hanya setengah jadi atau sepenuhnya jadi. Metode pengujian dapat dibagi menjadi 2, metode pengujian manual dan metode pengujian otomatis. Metode pengujian manual merupakan metode dimana pengujian dilakukan oleh sebuah user secara manual. Pengujian otomatis merupakan pengujian yang dilakukan dengan menggunakan sebuah alat untuk melakukan pengujian secara otomatis. Pada metode manual, diperlukan seorang *user* atau penguji untuk melakukan pengujian secara langsung. Penguji berinteraksi langsung dengan antarmuka aplikasi yang ada dan melakukan

pengujian terhadap fungsi melalui interaksi tersebut. Sedangkan pada pengujian otomatis, penguji tidak berinteraksi langsung dengan antarmuka aplikasi, melainkan menggunakan pihak ketiga seperti perangkat lunak penguji. Dalam pengujian otomatis, penguji akan mendesain alur pengujian pada perangkat lunak tersebut, dan perangkat lunak itulah yang nantinya akan berinteraksi dengan aplikasi saat pengujian.

3.4. Katalon Studio

Wikipedia menjelaskan bahwa Katalon studio merupakan sebuah perangkat pengujian yang dikembangkan oleh Katalon LLC. Perangkat lunak ini dibuat dengan menggunakan Selenium dan Appium sebagai dasarnya. Dirilis pada tahun 2016, pada tahun 2018 Katalon mendapatkan 9% dari penjualan di pasar untuk otomasi pengujian UI. Katalon menyediakan beberapa antarmuka untuk pengujian. Yang pertama adalah metode *record and play*, di mana pengguna dapat merekam langkah – langkah pengujian dan menjalankan langkah – langkah tersebut. Metode yang kedua adalah metode *scripting*. Metode ini lebih dikhususkan untuk para pengguna yang sudah ahli dan berpengalaman di bidang pemrograman. Selain itu Katalon juga memiliki plugin yang memudahkan pekerjaan seperti Git, Jira, Slack, dll. Perangkat lunak ini dapat digunakan untuk melakukan pengujian baik terhadap perangkat *mobile* ataupun halaman web [22]. Katalon Studio juga dapat melakukan pengujian terhadap perangkat lunak pada komputer atau API.

3.5. Black-Box Testing

Black-box Testing merupakan salah satu metode yang digunakan untuk melakukan pengujian terhadap perangkat lunak atau aplikasi. Suatu program atau sistem yang akan diuji dilihat sebagai sebuah *black-box*. Pengujian *black-box* dilakukan berdasarkan spesifikasi kebutuhan dan tidak perlu memeriksa kode pada sistem atau program tersebut. Pengujian *black-box* juga memiliki peran penting dalam pengujian perangkat lunak, karena pengujian ini membantu dalam validasi fungsionalitas keseluruhan sistem atau program. Keuntungan dari pengujian *black-box* adalah penguji tidak perlu memahami suatu bahasa pemrograman khusus, dan

penguji dan *programmer* tidak saling bergantung satu sama lain. Selain itu pengujian ini juga dianggap dilakukan dari sudut pandang pengguna. Pengujian *black-box* juga membantu untuk menemukan keanehan atau sesuatu yang tidak konsisten dalam spesifikasi kebutuhan [23]. *Black-box testing* juga memiliki beragam teknik dan beberapa di antaranya adalah seperti *state transition testing* yang menggunakan *input*, *output*, dan kondisi dari aplikasi ketika pengujian dan mengamati perubahan dalam aplikasi atau sistem, jika *output* dari hasil pengujian tersebut sesuai dengan yang diinginkan, maka berarti aplikasi tidak memiliki bug. Pada State Transition Testing dilakukan positif dan negatif *testing* untuk menguji kondisi normal ketika input yang dimasukkan sesuai dan kondisi alternative ketika input yang dimasukkan tidak sesuai format atau syarat pada aplikasi. *Decision table testing* menggunakan sebuah tabel berisi *test case* yang berdasar dari kemungkinan – kemungkinan yang ada dan setiap kondisi di cek dan dilakukan untuk mendapatkan *output* yang akurat. Selain itu, *Decision Table Testing* berisi serangkaian langkah – langkah dan kondisi yang diinputkan dan hasil akhir yang seharusnya didapatkan, Selain itu, pendekatan yang digunakan dalam black-box testing juga dapat dibagi menjadi 2 secara garis besar, yaitu *exploratory testing* dimana penguji melakukan pengujian secara langsung dan melakukan penilaian secara langsung, dan *scripted testing* dimana penguji melakukan pengujian berdasarkan skrip yang berisi langkah – langkah pengujian dan *test case* yang telah disiapkan sebelumnya. Metode yang akan digunakan adalah *state transition testing* dengan *scripted testing*.

BAB VI. PENUTUP

6.1. Kesimpulan

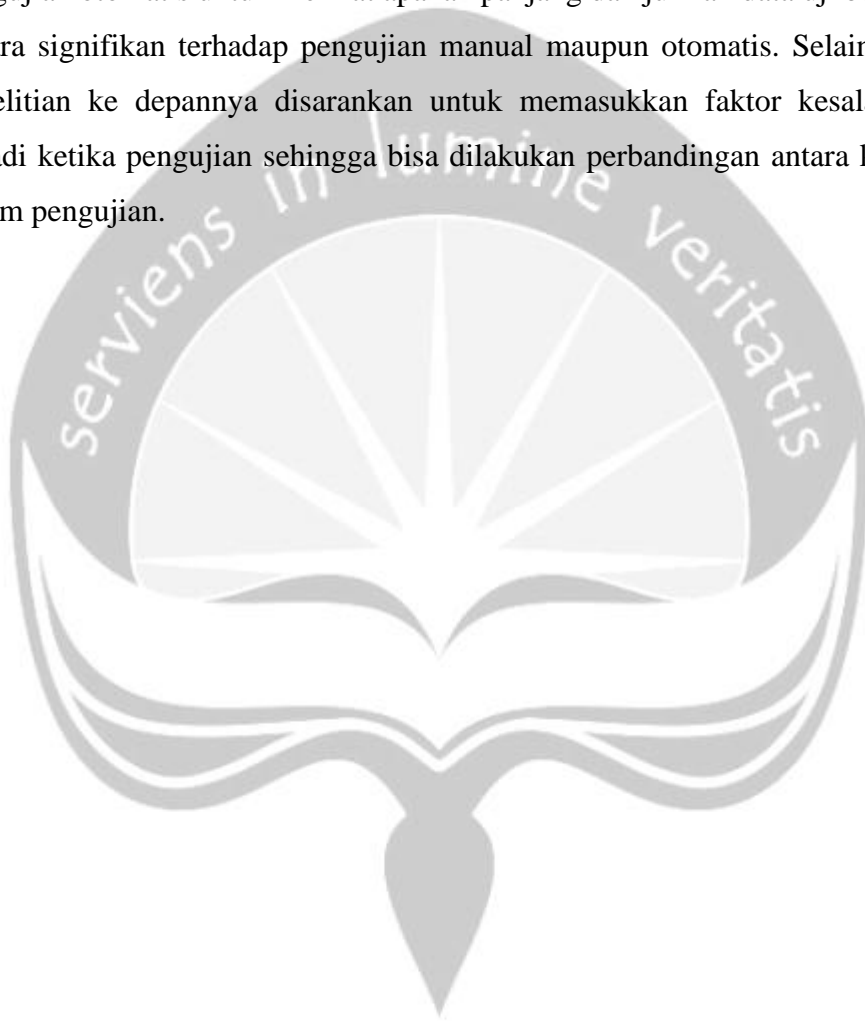
Beberapa kesimpulan yang dapat ditarik berdasarkan penelitian ini adalah sebagai berikut:

1. Cara melakukan pengujian manual adalah dengan penguji melakukan interaksi langsung dan memasukkan nilai – nilai tertentu secara manual ke dalam aplikasi melalui serangkaian aksi. Kemudian penguji harus menentukan kesesuaian antara fungsi dan alur yang seharusnya terjadi. Sementara pada pengujian otomatis. Penguji pertama harus merancang *test case* dalam Katalon Studio, membuat variabel – variabel yang diperlukan untuk menampung nilai dari data yang telah dibuat di excel yang dimasukkan ke Katalon Studio. Setelah itu, penguji hanya perlu menjalankan *test case* yang dimasukkan ke dalam *test suite* untuk menguji aplikasi secara otomatis tanpa interaksi terhadap aplikasi.
2. Perbandingan yang didapatkan setelah melakukan kedua pengujian secara otomatis dan manual adalah bahwa pengujian otomatis cocok digunakan untuk pengujian yang sudah memiliki spesifikasi yang jelas, sementara pengujian manual cocok untuk digunakan pada aplikasi yang sebelumnya belum pernah diuji. Pengujian otomatis dengan Katalon Studio menggunakan waktu yang lebih signifikan dengan rata – rata selisih waktu sebesar 61,47%. Hal ini disebabkan salah satu faktor signifikan yaitu *start application* dan waktu yang digunakan untuk langkah – langkah lainnya. Pengujian otomatis cocok digunakan untuk pengujian yang repetitive karena *test case* dapat digunakan untuk menguji ulang fitur sebelumnya jika ada tambahan fitur. Kemudian jumlah *bug* yang ditemukan adalah 10 *bug* dari 11 fungsi yang diuji.

6.2. Saran

Berdasarkan penelitian yang telah dilakukan di atas, penulis ingin menyampaikan saran menjadi pertimbangan dalam melakukan penelitian yang berkaitan dengan pengujian otomatis ke depannya, diantaranya yaitu;

Pada penelitian berikutnya, dapat dilakukan penambahan data uji ketika melakukan pengujian otomatis untuk melihat apakah panjang dan jumlah data uji berpengaruh secara signifikan terhadap pengujian manual maupun otomatis. Selain itu untuk penelitian ke depannya disarankan untuk memasukkan faktor kesalahan yang terjadi ketika pengujian sehingga bisa dilakukan perbandingan antara keefektifan dalam pengujian.



DAFTAR PUSTAKA

- [1] S. Singh, et al., "Automated Testing of Mobile Applications using Scripting Technique: A Study on Appium," *International Journal of Current Engineering and Technology* Vol. 4, No. 15, 2014.
- [2] H. Anjum, et al., "A Comparative Analysis of Quality Assurance of Mobile Applications using Automated Testing Tools," (*IJACSA*) *International Journal of Advanced Computer Science and Applications*, Vol. 8, No.7, 2017.
- [3] Trivedi, P., dan Pachori, S., "Modelling and Analysis of Software Defects Prevention Using ODC", *International Journal of Advanced Computer Science and Applications (IJACSA)*, Vol. 1, No. 3, pp75-77, 2010.
- [4] I. Burnstein, "Practical Software Testing: Process Oriented Approach, Springer Professional Computing, 2003.
- [5] Selvam, R, "Mobile Software Testing – Automated Test Case Design Strategies. *International Journal on Computer Science and Engineering*" (*IJCSE*) Vol.3, 2011.
- [6] Freeman, H. (2002). "Software Testing". *IEEE Instrumentation & Measurement Magazine*, P.48-50.
- [7] K. Karhu, T. Repo dan K. Smolander, "Empirical Observations on Software Testing Automation," *International Conference on Software Testing Verification and Validation*, (2009).
- [8] B. M. Amen, et al., "Mobile Application Testing Matrix and Challenges," *Fifth International conference on Computer Science and Information Technology*, At Sydney, Australia, Volume: 5, 2015.
- [9] G. J. Myers, T. Badgett, dan C. Sandler, "The Art of Software Testing," 3rd ed. USA: John Wiley & Sons, 2011.
- [10] Lovreto, Gabriel, et al, "Automated tests for Mobile Games: An Experience Report," *XVII Brazilian Symposium on Computer Games and Digital Entertainment (SBGAMES)*, At Foz do Iguaçu, 2018.
- [11] K. Valliammai dan Dr.P. Sujatha, "Analysis of Efficiency of Automated Software Testing Methods: Direction of Research," *International Journal of Science and Research (IJSR)*, 2015.
- [12] C. Karner, "Exploratory Testing," *Quality Assurance Institute Worldwide Annual Software Testing Conference*, Orlando, FL, 2014.
- [13] H. Muccini, et al., "Software testing of mobile applications: Challenges and future research directions," *Automation of Software Test (AST)*, 7th International Workshop on , vol., no., pp.29,35, 2012
- [14] A. Kaur, "Review of Mobile Applications Testing with Automated Techniques." *International Journal of Advanced Research in Computer and Communication Engineering* Vol. 4, Issue 10, October 2015
- [15] Zhongqian, et al., "Keyword-Driven Testing Framework For Android

- Applications." Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)
- [16] K. Haengkon, "Test Driven Mobile Applications Development." Proceedings of the World Congress on Engineering and Computer Science. Vol. 2. 2013.
- [17] D. Amalfitano, et al., "A GUI Crawling-Based Technique for Android Mobile Application Testing." Software Testing, Verification and Validation Workshops (ICSTW), IEEE Fourth International Conference on, pp. 252-261, 2011.
- [18] L. Nagowah, et al., "A novel approach of automation testing on mobile devices," Computer & Information Science (ICCIS), 2012 International Conference on , vol.2, no., pp.924,930, 12-14 June 2012
- [19] M. Ilen, "Bug Tracking Basics: A beginner's guide to reporting and tracking defects". Software Testing & Quality Engineering Magazine. Vol. 4 no. 3. pp. 20–24, 2017.
- [20] C. Sharma et al., "A Survey on Software Testing Techniques using Genetic Algorithm", International Journal of Computer Science Issues, Vol. 10, Issue 1, No 1, January 2013.
- [21] A. Kolawa dan D. Huizinga, "Automated Defect Prevention: Best Practices in Software Management," Wiley-IEEE Computer Society Press, 2007.
- [22] <https://www.katalon.com/katalon-studio/>
- [23] S. Nidhra dan J. Dondeti., "Black-Box and White-Box Testing Techniques-A Literature Review," International Journal of Embedded Systems and Applications (IJESA) Vol.2, No.2, 2012