

***FUNCTIONAL TESTING PADA APLIKASI MOBILE
ACC.ONE***

Tugas Akhir

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



Dibuat Oleh:

RIZKA YULIANTI PRATIWI

16 07 08926

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

HALAMAN PENGESAHAN

Tugas Akhir Berjudul

FUNCTIONAL TESTING PADA APLIKASI MOBILE ACC.ONE

yang disusun oleh

RIZKA YULIANTI PRATIWI

160708926

dinyatakan telah memenuhi syarat pada tanggal 16 April 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, 16 April 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 : Rizka Yulianti Pratiwi
NPM : 16 07 08926
Program Studi : Informatika
Fakultas : Teknologi Industri
Judul Penelitian : *Functional Testing* Pada Aplikasi *Mobile acc.one*

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, 16 April 2020

Yang menyatakan,

Rizka Yulianti Pratiwi

16 07 08926

PERNYATAAN PERSETUJUAN DARI INSTANSI

ASAL PENELITIAN

Saya yang bertanda tangan di bawah ini:

Nama Lengkap Pembimbing : Kevin Antariksa

Jabatan : *IT System Development Analyst Programmer & Coordinator Technocentre Yogyakarta*

Departemen : *IT Solution*

Divisi : *IT*

Menyatakan dengan ini:

Nama Lengkap : Rizka Yulianti Pratiwi

NPM : 16 07 08926

Program Studi : Informatika

Fakultas : Teknologi Industri

Judul Penelitian : *Functional Testing Pada Aplikasi Mobile acc.one*

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, 16 April 2020

Yang menyatakan,

Kevin Antariksa
*IT System Development
Analyst Programmer &
Coordinator Technocentre
Yogyakarta*

HALAMAN PERSEMBAHAN

وَلَقَدْ ءَاتَيْنَا لُقْمَانَ الْحِكْمَةَ أَنِ اشْكُرْ لِلَّهِ وَمَن يَشْكُرْ فَإِنَّمَا يَشْكُرُ لِنَفْسِهِ ۗ
وَمَن كَفَرَ فَإِنَّ اللَّهَ غَنِيٌّ حَمِيدٌ ﴿١٢﴾

*And We had certainly given Luqman wisdom [and said], "Be grateful to Allah"
And whoever is grateful is grateful for [the benefit of] himself. And whoever
denies [His favor] - then indeed, Allah is Free of need and Praiseworthy.*

[Luqman 31:12]



*“Terima kasih teruntuk kotaku Yogyakarta, atas semua rasa kenanganku dan
teman-teman semasa kuliah, ku meminta izin tuk menitipkannya disini, aku
selalu berharap kepada Tuhan agar suatu saat nanti aku akan kembali pulang
ke kotamu. Kupercaya selalu ada sesuatu di Jogja. Amin.”*

- Yogyakarta, 16 April 2020 -

KATA PENGANTAR

Puji syukur dan terima kasih senantiasa penulis panjatkan kepada Tuhan Yang Maha Esa karena Ia selalu senantiasa memberikan nikmat, kasih sayang, dan karunia sehingga penulis dapat menyelesaikan penulisan tugas akhir ini. Rasa hormat dan terima kasih penulis ucapkan juga kepada:

1. Allah SWT, Yang Maha Pengasih lagi Maha Penyayang, dengan limpahan nikmat, kasih sayang, dan karunia-Nya
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. Pihak Astra Credit Companies yang telah membantu penulis selama magang dan mengerjakan tugas akhir.
6. Mama, Papa, dan keluarga lainnya yang hingga kini mendukung penulis.
7. Yokebeth Denna Leksanti, Geralda Ken Sadhani, Mawar Listra Theresa, Yulita Flaviana Bahy, Agri Nuhgraha, Hugo Candraditya, Devina Arnyndiasari, dan Azarya Abednego yang telah menjadi periang dan teman terdekat selama masa perkuliahan.
8. Lukito Budi Prasetyo yang telah mendampingi 3 tahun awal masa perkuliahan dan menjadi salah satu orang terdekat yang spesial bagi penulis.
9. Teman-teman Informatika 2016 Kelas C, Kelompok Studi Pemrograman (KSP), dan Redaksi IMAGE, yang telah membantu penulis berproses dalam pengembangan *soft skill* sekaligus menjadi keluarga baru selama penulis duduk di bangku perkuliahan.
10. Teman-teman Asisten Praktikum Sistem Digital 2018 & 2019, Arsitektur Komputer 2019, Pengembangan Berbasis Platform 2019, yang telah menjadi

keluarga baru dan tempat berbagi dalam mengembangkan *soft skill*.

11. Teman tim P3L, yaitu Hendrikus Adi Purnama dan Wilson Saputra atas kerja sama yang sangat menyenangkan.
12. Teman-teman magang di ACC *Batch* 3 dan 4: Ryandi, Hugo, Frentzen, Theo, Geo, Leo, Setiawan, Alda, Denna, Fityan, Intan, Christi, Marvin, dan Tono yang menemani hari-hari ketika magang.
13. Diah Intan Pramitha yang merupakan teman magang yang telah membantu menemani, berbagi, dan menjadi teman curhat bagi penulis.
14. Kak Tara dan Kak Kevin yang telah membantu membimbing penulis selama mengerjakan tugas magang dan tugas akhir.

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

Yogyakarta, 16 April 2020

Rizka Yulianti Pratiwi

16 07 08926

DAFTAR ISI

JUDUL	i
HALAMAN PENGESAHAN.....	ii
PERNYATAAN ORISINALITAS & PUBLIKASI ILMIAH.....	iii
PERNYATAAN PERSETUJUAN DARI INSTANSI ASAL PENELITIAN..	iv
HALAMAN PERSEMBAHAN	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	i
DAFTAR GAMBAR.....	v
DAFTAR TABEL.....	xi
INTISARI.....	xii
BAB I. PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	4
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	4
1.5 Metode Penelitian	4
1.6 Sistematika Penulisan	6
BAB II. TINJAUAN PUSTAKA.....	7
2.1 Penelitian Terdahulu	7
BAB III. LANDASAN TEORI.....	14
3.1 Pengujian Perangkat Lunak	14
a. <i>Blackbox Testing</i>	15
b. <i>Whitebox Testing</i>	16
3.2 Aplikasi <i>Mobile acc.one</i>	17

3.3	Katalon Studio	18
BAB IV. ANALISIS DAN PERANCANGAN PENGUJIAN TEST CASE....		21
4.1	Deskripsi Masalah.....	21
4.2	Analisis Kebutuhan Pengujian.....	21
4.2.1	Analisis Kebutuhan Pengujian Manual	21
4.2.2	Analisis Kebutuhan Pengujian Otomatis.....	21
4.3	Perancangan Pengujian	22
4.3.1	Perancangan Tujuan	22
4.3.2	Perancangan <i>Test Case</i>	22
4.3.2.1	<i>Use Case</i> Mencari Mobil Baru	24
4.3.2.2	<i>Use Case</i> Mencari Promo Mobil Tertentu.....	29
4.3.2.3	<i>Use Case</i> Mencari Mobil Bekas (<i>Marketplace</i>)	32
4.3.2.4	<i>Use Case</i> Mencari Mobil Bekas (Lelang)	37
4.3.2.5	<i>Use Case</i> Mengubah Foto Profil	43
4.3.2.6	<i>Use Case</i> Mencari Mobil Rental	47
4.3.2.7	<i>Use Case</i> Lupa Kata Sandi	56
4.3.2.8	<i>Use Case</i> Melihat Status Pengajuan Mobil	71
4.3.2.9	<i>Use Case</i> Simulasi Kredit Mobil Baru (Paket).....	75
4.3.2.10	<i>Use Case</i> Melihat Paket Pembelian Mobil.....	81
BAB V. HASIL EKSPERIMEN DAN PEMBAHASAN		84
5.1	Deskripsi Pengujian	84
5.1.1	Alur Pengujian Secara Manual.....	84
5.1.2	Alur Pengujian Secara Otomatis	84
5.2	Hasil Pengujian	86
5.2.1	Analisis Hasil Pengujian Manual	86

5.2.1.1	Penerapan <i>Manual Testing</i> Untuk <i>Use Case</i> Mencari Mobil Baru .	86
5.2.1.2	Penerapan <i>Manual Testing</i> Untuk <i>Use Case</i> Mencari Promo Mobil Tertentu	89
5.2.1.3	Penerapan <i>Manual Testing</i> Untuk <i>Use Case</i> Mencari Mobil Bekas (<i>Marketplace</i>)	90
5.2.1.4	Penerapan <i>Manual Testing</i> Untuk <i>Use Case</i> Mencari Mobil Bekas (Lelang)	93
5.2.1.5	Penerapan <i>Manual Testing</i> Untuk <i>Use Case</i> Mengubah Foto Profil	96
5.2.1.6	Penerapan <i>Manual Testing</i> Untuk <i>Use Case</i> Mencari Mobil Rental	97
5.2.1.7	Penerapan <i>Manual Testing</i> Untuk <i>Use Case</i> Lupa Kata Sandi	100
5.2.1.8	Penerapan <i>Manual Testing</i> Untuk <i>Use Case</i> Melihat Status Pengajuan Mobil	118
5.2.1.9	Penerapan <i>Manual Testing</i> Untuk <i>Use Case</i> Simulasi Kredit Mobil Baru (Paket)	120
5.2.1.10	Penerapan <i>Manual Testing</i> Untuk <i>Use Case</i> Melihat Paket Pembelian Mobil	122
5.2.2	Analisis Hasil Pengujian Otomatis	123
5.2.2.1	Penerapan <i>Automated Testing</i> Pada Katalon Untuk <i>Use Case</i> Mencari Mobil Baru	123
5.2.2.2	Penerapan <i>Automated Testing</i> Pada Katalon Untuk Untuk <i>Use Case</i> Mencari Promo Mobil Tertentu	130
5.2.2.3	Penerapan <i>Automated Testing</i> Pada Katalon Untuk Untuk <i>Use Case</i> Mencari Mobil Bekas (<i>Marketplace</i>)	134
5.2.2.4	Penerapan <i>Automated Testing</i> Pada Katalon Untuk <i>Use Case</i> Mencari Mobil Bekas (Lelang)	142

5.2.2.5	Penerapan <i>Automated Testing</i> Pada Katalon Untuk <i>Use Case</i> Mengubah Foto Profil	147
5.2.2.6	Penerapan <i>Automated Testing</i> Pada Katalon Untuk <i>Use Case</i> Mencari Mobil Rental	151
5.2.2.7	Penerapan <i>Automated Testing</i> Pada Katalon Untuk <i>Use Case</i> Lupa Kata Sandi	159
5.2.2.8	Penerapan <i>Automated Testing</i> Pada Katalon Untuk <i>Use Case</i> Melihat Status Pengajuan Mobil	166
5.2.2.9	Penerapan <i>Automated Testing</i> Pada Katalon Untuk <i>Use Case</i> Simulasi Kredit Mobil Baru (Paket)	170
5.2.2.10	Penerapan <i>Automated Testing</i> Pada Katalon Untuk <i>Use Case</i> Melihat Paket Pembelian Mobil.....	175
5.3	Pembahasan Hasil Perbandingan Pengujian Manual dan Otomatis	178
BAB VI. PENUTUP		183
6.1	Kesimpulan	183
6.2	Saran	183
DAFTAR PUSTAKA		185

DAFTAR GAMBAR

Gambar 1. Grafik Pengguna Smartphone di Indonesia 2016-2019	1
Gambar 2. Aplikasi acc.one dan Fiturnya.....	17
Gambar 3. Pengaturan Mode Pengembang.....	19
Gambar 4. Perangkat Pengujian yang Sudah Tersambung Dengan Komputer. ...	20
Gambar 5. Use Case Diagram Aplikasi Mobile acc.one.....	22
Gambar 6. Bagan Alur Use Case Mencari Mobil Baru	26
Gambar 7. Bagan Alur Use Case Mencari Promo Tertentu.....	30
Gambar 8. Bagan Alur Use Case Mencari Mobil Bekas (Marketplace).....	34
Gambar 9. Bagan Alur Use Case Mencari Mobil Bekas (Lelang).....	39
Gambar 10. Bagan Alur Use Case Mengubah Foto Profil.....	45
Gambar 11. Bagan Alur Use Case Mencari Mobil Rental.....	49
Gambar 12. Bagan Alur Use Case Lupa Kata Sandi	59
Gambar 13. Bagan Alur Use Case Melihat Status Pengajuan Mobil.....	73
Gambar 14. Bagan Alur Use Case Simulasi Kredit Mobil Baru Paket.....	77
Gambar 15. Bagan Alur Use Case Melihat Paket Pembelian Mobil	82
Gambar 16. Bagan Alur Pengujian acc.one Secara Manual.	84
Gambar 17. Bagan Alur Pengujian acc.one Secara Otomatis.....	85
Gambar 18. Test Case Katalon Untuk Use Case Mencari Mobil Baru (1).....	123
Gambar 19. Test Case Katalon Untuk Use Case Mencari Mobil Baru (2).....	123
Gambar 20. Test Case Katalon Untuk Use Case Mencari Mobil Baru (3).....	123
Gambar 21. Kustomisasi Pembuatan Objek Baru.....	124
Gambar 22. Salah Satu Contoh Penggunaan Objek Baru Pada Variabel Spesific.	124
Gambar 23. Variabel Katalon Untuk Use Case Mencari Mobil Baru.....	126
Gambar 24. Test Suite Katalon Untuk Use Case Mencari Mobil Baru.	126
Gambar 25. Data Binding Untuk Use Case Mencari Mobil Baru.....	127
Gambar 26. Hasil Pengujian Automated Testing Untuk Use Case Mencari Mobil Baru (1).	128
Gambar 27. Hasil Pengujian Automated Testing Untuk Use Case Mencari Mobil Baru (2).	128

Gambar 28. Hasil Pengujian Automated Testing Untuk Use Case Mencari Mobil Baru (3).	129
Gambar 29. Test Case Katalon Untuk Use Case Mencari Promo Mobil Tertentu.	130
Gambar 30. Variabel Katalon Untuk Use Case Mencari Promo Mobil Tertentu.	130
Gambar 31. Test Suite Katalon Untuk Use Case Mencari Promo Mobil Tertentu.	131
Gambar 32. Data Binding Untuk <i>Use Case</i> Mencari Promo Mobil Tertentu.....	131
Gambar 33. Hasil Pengujian Automated Testing Untuk Use Case Mencari Promo Mobil Tertentu (1).	132
Gambar 34. Hasil Pengujian Automated Testing Untuk Use Case Mencari Promo Mobil Tertentu (2).	132
Gambar 35. Hasil Pengujian Automated Testing Untuk Use Case Mencari Promo Mobil Tertentu (3).	132
Gambar 36. Test Case Katalon Untuk Use Case Mencari Mobil Bekas (Marketplace) (1).	134
Gambar 37. Test Case Katalon Untuk Use Case Mencari Mobil Bekas (Marketplace) (2).	134
Gambar 38. Variabel Katalon Untuk Use Case Mencari Mobil Bekas (Marketplace).	135
Gambar 39. Test Suite Katalon Untuk Use Case Mencari Mobil Bekas (Marketplace).	136
Gambar 40. Data Binding Untuk Use Case Mencari Mobil Bekas (Marketplace).	136
Gambar 41. Hasil Automated Testing Untuk Use Case Mencari Mobil Bekas (Marketplace) (1).	137
Gambar 42. Hasil Automated Testing Untuk Use Case Mencari Mobil Bekas (Marketplace) (2).	138
Gambar 43. Hasil Automated Testing Untuk Use Case Mencari Mobil Bekas (Marketplace) (3).	139

Gambar 44. Hasil Automated Testing Untuk Use Case Mencari Mobil Bekas (Marketplace) (4).....	139
Gambar 45. Bug yang Ditemukan Pada Use Case Mencari Mobil Bekas (Marketplace) Brand Toyota.	140
Gambar 46. Bug yang Ditemukan Pada Use Case Mencari Mobil Bekas (Marketplace) Brand Daihatsu.	140
Gambar 47. Test Case Katalon Untuk Use Case Mencari Mobil Bekas (Lelang) (1).	142
Gambar 48. Test Case Katalon Untuk Use Case Mencari Mobil Bekas (Lelang) (2).	142
Gambar 49. Variabel Katalon Untuk Use Case Mencari Mobil Bekas (Lelang).	143
Gambar 50. Test Suite Katalon Untuk Use Case Mencari Mobil Bekas (Lelang).	144
Gambar 51. Data Binding Untuk Use Case Mencari Mobil Bekas (Lelang).	144
Gambar 52. Hasil Automated Testing Untuk Use Case Mencari Mobil Bekas (Lelang) (1).	145
Gambar 53. Hasil Automated Testing Untuk Use Case Mencari Mobil Bekas (Lelang) (2).	145
Gambar 54. Hasil Automated Testing Untuk Use Case Mencari Mobil Bekas (Lelang) (3).	146
Gambar 55. Test Case Katalon Untuk Use Case Mengubah Foto Profil.	147
Gambar 56. Variabel Katalon Untuk Use Case Mengubah Foto Profil.	147
Gambar 57. Test Suite Katalon Untuk Use Case Mengubah Foto Profil.	148
Gambar 58. Data Binding Untuk Use Case Mengubah Foto Profil.	148
Gambar 59. Hasil Automated Testing Untuk Use Case Mengubah Foto Profil (1).	149
Gambar 60. Hasil Automated Testing Untuk Use Case Mengubah Foto Profil (2).	149
Gambar 61. Hasil Automated Testing Untuk Use Case Mengubah Foto Profil (3).	150
Gambar 62. Test Case Katalon Untuk Use Case Mencari Mobil Rental (1).	151
Gambar 63. Test Case Katalon Untuk Use Case Mencari Mobil Rental (2).	151

Gambar 64. Test Case Katalon Untuk Use Case Mencari Mobil Rental (3).	152
Gambar 65. Test Case Katalon Untuk Use Case Mencari Mobil Rental (4).	152
Gambar 66. Test Case Katalon Untuk Use Case Mencari Mobil Rental (5).	153
Gambar 67. Test Case Katalon Untuk Use Case Mencari Mobil Rental (6).	153
Gambar 68. Test Case Katalon Untuk Use Case Mencari Mobil Rental (7).	154
Gambar 69. Variabel Katalon Untuk Use Case Mencari Mobil Rental.	155
Gambar 70. Test Suite Katalon Untuk Use Case Mencari Mobil Rental.	155
Gambar 71. Data Binding Untuk Use Case Mencari Mobil Rental (1).	156
Gambar 72. Data Binding Untuk Use Case Mencari Mobil Rental (2).	156
Gambar 73. Data Binding Untuk Use Case Mencari Mobil Rental (3).	156
Gambar 74. Hasil Automated Testing Untuk Use Case Mencari Mobil Rental (1).	157
Gambar 75. Hasil Automated Testing Untuk Use Case Mencari Mobil Rental (2).	157
Gambar 76. Hasil Automated Testing Untuk Use Case Mencari Mobil Rental (3).	157
Gambar 77. Test Case Katalon Untuk Use Case Lupa Kata Sandi (1).	159
Gambar 78. Test Case Katalon Untuk Use Case Lupa Kata Sandi (2).	159
Gambar 79. Test Case Katalon Untuk Use Case Lupa Kata Sandi (3).	160
Gambar 80. Variabel Katalon Untuk Use Case Lupa Kata Sandi.	161
Gambar 81. Test Suite Katalon Untuk Use Case Lupa Kata Sandi.	161
Gambar 82. Data Binding Untuk Use Case Lupa Kata Sandi.	162
Gambar 83. Hasil Automated Testing Untuk Use Case Lupa Kata Sandi (1). ...	163
Gambar 84. Hasil Automated Testing Untuk Use Case Lupa Kata Sandi (2). ...	164
Gambar 85. Hasil Automated Testing Untuk Use Case Lupa Kata Sandi (3). ...	164
Gambar 86. Bug yang Ditemukan Pada Use Case Lupa Kata Sandi.	165
Gambar 87. Test Case Katalon Untuk Use Case Melihat Status Pengajuan Mobil.	166
Gambar 88. Variabel Katalon Untuk Use Case Melihat Status Pengajuan Mobil.	166
Gambar 89. Test Suite Katalon Untuk Use Case Melihat Status Pengajuan Mobil.	167

Gambar 90. Data Binding Untuk Use Case Melihat Status Pengajuan Mobil....	167
Gambar 91. Hasil Automated Testing Untuk Use Case Melihat Status Pengajuan Mobil (1).	168
Gambar 92. Hasil Automated Testing Untuk Use Case Melihat Status Pengajuan Mobil (2).	168
Gambar 93. Hasil Automated Testing Untuk Use Case Melihat Status Pengajuan Mobil (3).	169
Gambar 94. Test Case Katalon Untuk Use Case Simulasi Kredit Mobil Baru (Paket) (1).....	170
Gambar 95. Test Case Katalon Untuk Use Case Simulasi Kredit Mobil Baru (Paket) (2).....	170
Gambar 96. Variabel Katalon Untuk Use Case Simulasi Kredit Mobil Baru (Paket).	171
Gambar 97. Test Suite Katalon Untuk Use Case Simulasi Kredit Mobil Baru (Paket).	172
Gambar 98. Data Binding Untuk Use Case Simulasi Kredit Mobil Baru (Paket).	172
Gambar 99. Hasil Automated Testing Untuk Use Case Simulasi Kredit Mobil Baru (Paket) (1).....	173
Gambar 100. Hasil Automated Testing Untuk Use Case Simulasi Kredit Mobil Baru (Paket) (2).....	173
Gambar 101. Hasil Automated Testing Untuk Use Case Simulasi Kredit Mobil Baru (Paket) (3).....	174
Gambar 102. Test Case Katalon Untuk Use Case Melihat Paket Pembelian Mobil.	175
Gambar 103. Variabel Katalon Untuk Use Case Melihat Paket Pembelian Mobil.	175
Gambar 104. Hasil Automated Testing Untuk Use Case Melihat Paket Pembelian Mobil (1).	176
Gambar 105. Hasil Automated Testing Untuk Use Case Melihat Paket Pembelian Mobil (2).	176

Gambar 106. Hasil Automated Testing Untuk Use Case Melihat Paket Pembelian Mobil (3).	177
Gambar 107. Custom Keyword Untuk Scroll Up Down.	178
Gambar 108. Salah Satu Contoh Laporan yang Di Generate Secara Otomatis Oleh Katalon Setelah Pengujian.	182



DAFTAR TABEL

Tabel 1. Perbandingan Penelitian.....	12
Tabel 2. Nilai Aktual Test Case Untuk Use Case Mencari Mobil Baru.	86
Tabel 3. Nilai Aktual Test Case Untuk Use Case Mencari Promo Mobil Tertentu.	89
Tabel 4. Nilai Aktual Test Case Untuk Use Case Mencari Mobil Bekas (Marketplace).....	90
Tabel 5. Nilai Aktual Test Case Untuk Use Case Mencari Mobil Bekas (Lelang).	93
Tabel 6. Nilai Aktual Test Case Untuk Use Case Mengubah Foto Profil.....	96
Tabel 7. Nilai Aktual Test Case Untuk Use Case Mencari Mobil Rental.....	97
Tabel 8. Nilai Aktual Test Case Untuk Use Case Lupa Kata Sandi.	101
Tabel 9. Nilai Aktual Test Case Untuk Use Case Melihat Status Pengajuan Mobil.	118
Tabel 10. Nilai Aktual Test Case Untuk Use Case Simulasi Kredit Mobil Baru (Paket).....	120
Tabel 11. Nilai Aktual Test Case Untuk Use Case Melihat Paket Pembelian Mobil.....	122
Tabel 12. Perbandingan Waktu Antara Pengujian Manual Dan Pengujian Otomatis.....	179

INTISARI

FUNCTIONAL TESTING PADA APLIKASI MOBILE ACC.ONE

Rizka Yulianti Pratiwi

16 07 08926

Penggunaan *smartphone* yang meningkat dalam beberapa tahun terakhir membuat para *developer* terpacu untuk membuat aplikasi *mobile* yang dapat bermanfaat dalam kehidupan sehari-hari. Tak terkecuali PT. Astra Sedaya Finance yang meluncurkan aplikasi *mobile acc.one* dengan tujuan untuk membantu para *customer* dalam pelayanan pembelian hingga lelang mobil baru maupun bekas secara *online*, serta menjawab pertanyaan *customer* menggunakan sistem *bot chat* di mana pun dan kapan pun. Namun, hingga kini aplikasi yang akan menjadi *single point of contact* bagi para pelanggannya tersebut masih dalam tahap pengembangan. Maka dari itu, perlu dilakukan pengujian seiring perkembangannya, agar aplikasi *acc.one* memiliki fungsi yang sesuai dengan spesifikasi awal *user* sehingga dapat menjadikan aplikasi tersebut berkualitas dan nyaman digunakan oleh pelanggannya.

Pada tugas akhir ini akan dilakukan pengujian terhadap aplikasi *acc.one* secara manual dan otomatis. Dalam pengujian tersebut digunakan metode *blackbox* yang berarti *functional testing*. Fungsi yang akan diuji pada penelitian kali ini sebanyak 10 dari 33 fungsi yang ada. Pengujian ini nantinya akan membandingkan keefektifan pengujian secara manual dan otomatis. Dalam pengujian akan dicatat waktu yang dibutuhkan masing-masing fungsi baik secara manual maupun otomatis, serta pula akan dilaporkan kendala apa saja yang dihadapi pada saat pengujian.

Berdasarkan pengujian yang dilakukan terhadap aplikasi *mobile acc.one*, ditemukan *bug* di 2 dari 10 fungsi utama yang diuji. Setelah perhitungan dengan membandingkan kedua cara pengujian, didapatkan persentase sebesar 55,19% lebih cepat jika dilakukan secara manual. Namun, dibalik cepatnya pengujian secara manual, terdapat juga kelemahannya, seperti masih sering terjadinya kesalahan pada saat *input data*, sedangkan keunggulan pada pengujian otomatis terletak pada kemudahan dalam proses pelaporan dan eksekusi yang sudah jelas dilakukan secara auto oleh *tool uji*, sehingga tidak banyak melibatkan penguji dalam prosesnya. Kelemahan pada pengujian otomatis adalah banyaknya langkah yang dijalankan dikarenakan kebutuhan waktu untuk pendeteksian objek dan penentuan alur *test case*.

Kata kunci: *Functional Testing*, *Blackbox*, *acc.one*, Pengujian Otomatis, Pengujian Manual.

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

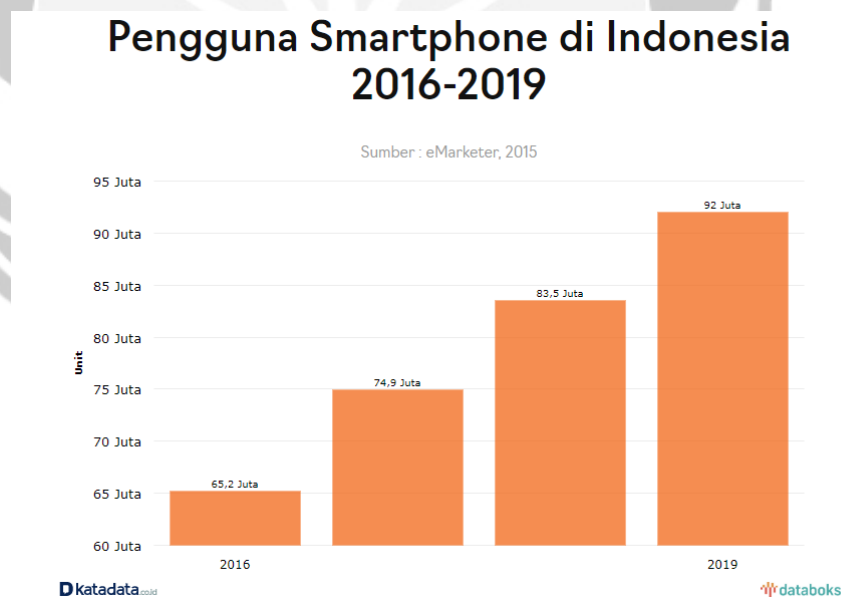
Dosen Pembimbing II : Stephanie Pamela Adithama S.T., M.T.

Jadwal Sidang Tugas Akhir : 16 April 2020

BAB I. PENDAHULUAN

1.1 Latar Belakang

Aplikasi berbasis *mobile* merupakan salah satu platform yang sedang digandrungi di era industri 4.0 ini. Pernyataan tersebut didukung dengan penggunaan *smartphone* yang semakin meningkat di setiap tahunnya di Indonesia. Berdasarkan eMarketer, tahun 2015 hingga 2019 peningkatan pengguna *smartphone* mencapai 26.8 juta pengguna dilihat berdasarkan data penjualan unit *smartphone* di Indonesia [1]. Menurut Gartner, pada tahun 2017 sendiri lebih dari 268 miliar aplikasi yang diunduh akan menghasilkan pendapatan kumulatif sebesar 77 miliar US Dollar. Maka dari itu, semakin banyak pengguna *smartphone* akan memengaruhi kualitas dari aplikasi yang diunduh yang di mana kualitas itu sendiri menjadi salah satu faktor penting dalam proses pembuatan aplikasi dan hal yang diutamakan sebelum aplikasi dirilis [2].



Gambar 1. Grafik Pengguna *Smartphone* di Indonesia 2016-2019 [1].

Dengan pengecekan aplikasi *mobile* sebelum didistribusikan secara luas melalui *smartphone*, maka dapat dilihat kualitas suatu aplikasi mengenai kelayakannya. Contohnya adalah *game testing* yang kebanyakan akan bergantung pada sumber daya manusia sebagai *tester*. Hal tersebut bukan

hanya akan memakan waktu dan pembiayaan dari sumber daya, akan tetapi berisiko pada potensi *error* dan tidak akuratnya hasil yang didapatkan dari manusia itu sendiri. Pernyataan ini diperkuat oleh pendapat dari Whittaker bahwa penerapan pengujian secara manual adalah tindakan padat karya dan rawan oleh kesalahan [3].

Untuk mengatasi banyaknya risiko yang ditimbulkan dari pengecekan manual, muncullah pengujian otomatis. Pada dasarnya, pengujian aplikasi dengan menggunakan pengujian otomatis berfungsi untuk mengurangi kebutuhan upaya manusia akan tetapi itu tidak berpengaruh pada kebutuhan untuk melakukan pengujian manual sama sekali, karena beberapa faktor dari aplikasi yang dapat diberi masukan hanya oleh manusia antara lain seperti tampilan atau desain aplikasi, serta pengalaman penggunaan aplikasi [4]. Namun untuk pengujian fungsional dengan menggunakan pengujian otomatis maka hasil yang didapatkan lebih akurat dan minim *human error* karena sumber daya utama dari pengecekan adalah sistem komputer. Ketika para *tester* telah mengembangkan sebuah skenario *test case* menggunakan pengujian otomatis untuk suatu *use case*, maka skenario tersebut dapat digunakan berulang, sehingga akan mengurangi waktu yang digunakan untuk pengecekan [5].

Pada penelitian ini akan dilakukan pengujian pada aplikasi acc.one milik PT. Astra Sedaya Finance di mana acc.one sendiri merupakan salah satu aplikasi dengan cakupan besar karena berfungsi sebagai integrasi dari seluruh bisnis yang dinaungi oleh ACC (Astra Credit Companies) yang berfungsi untuk mempermudah *customer* ACC untuk berinteraksi dan bertransaksi dengan perusahaan dan menjadikan *single point of contact* bagi *end customer* di mana aplikasi ini harus dapat melayani seluruh *customer* ACC pada layanan jasa maupun produk yang ditawarkan ACC di aplikasi acc.one, sehingga diharapkan nantinya *customer* akan nyaman menggunakan aplikasi *mobile* ini sebagai perantara. Aplikasi acc.one nantinya akan diuji secara fungsionalitas atau biasa disebut metode *blackbox* dengan maksud untuk mencari kesalahan dan menguji kesesuaian fungsi dengan spesifikasi *user*. Harapan penulis dengan adanya pengujian terhadap aplikasi acc.one

yang hingga kini masih dalam tahap pengembangan, dapat ditemukannya kesalahan sebanyak-banyaknya sebagai *feedback* untuk pengembang agar nantinya dapat dibenahi dan dapat menyesuaikan fungsi-fungsi pada aplikasi dengan spesifikasi awal yang diberikan oleh *user* agar *customer* tidak kecewa dengan layanan yang diberikan ACC di aplikasi acc.one.

Alur penelitian terhadap pengujian secara manual dan otomatis terhadap aplikasi acc.one nantinya kurang lebih akan sama. Alur pengujian akan dimulai dari menganalisis alur pengujian per fungsi secara manual atau melihat pada dokumentasi yang diberikan oleh perusahaan, kemudian penulis akan membuat *test case* sebagai salah satu syarat dan dasar dari dilakukannya pengujian (jika pengujian secara otomatis pembuatan *test case* akan dibuat di *tool* yang bersangkutan), lalu *functional testing* terhadap aplikasi *mobile* acc.one akan dilakukan secara manual dan otomatis, dan terakhir penulis akan membuat *output* laporan yang berisi keterangan bahwa apakah terdapat kesalahan pada fungsi-fungsi yang diuji atau tidak. Sedangkan laporan pada pengujian otomatis nantinya akan *generate* laporan secara otomatis setelah eksekusi.

Hingga kini, sudah banyak tersedia *frameworks* dan *tools* yang mendukung untuk pengujian aplikasi. Beberapa *tools* juga menyediakan *emulator* atau perangkat tiruan sebagai pengganti dari perangkat yang sebenarnya jika fasilitas kurang memadai, sebagai contoh yaitu Katalon dengan bahasa pemrograman yaitu Groovy. Beberapa *framework* hanya dapat menggunakan satu bahasa pemrograman, sedangkan sisanya dapat menggunakan bahasa yang lebih bervariasi seperti Java, JavaScript, maupun Python [6]. Digunakannya Katalon sebagai salah satu *tool* dalam pengujian otomatis acc.one bertujuan agar pengujian dan pengelolaan laporan lebih praktis, efisien, dan efektif karena di dalam Katalon sendiri semuanya sudah bekerja secara otomatis.

1.2 Rumusan Masalah

Berdasarkan latar belakang pada poin 1.1, maka diperoleh rumusan masalah sebagai berikut:

1. Bagaimana keunggulan dan kelemahan *functional testing* secara manual dan otomatis dengan *tool* uji Katalon untuk aplikasi *mobile acc.one*?
2. Bagaimana keberhasilan dari fungsi-fungsi yang diuji pada aplikasi *mobile acc.one*?

1.3 Batasan Masalah

Berdasarkan permasalahan di atas, maka penelitian ini akan dibatasi pada hal-hal sebagai berikut:

1. Pengujian aplikasi *mobile acc.one* hanya akan menguji dari sisi fungsionalitas.
2. Banyaknya fungsi yang diuji pada penelitian kali ini adalah sebanyak 10 dari 33 unit fungsi.
3. Sebagian fungsi diuji pada aplikasi *mobile acc.one* dengan versi *update* aplikasi per tanggal 21 Januari 2020 dan sebagian pada tanggal 12 Februari 2020.

1.4 Tujuan Penelitian

Penelitian ini memiliki tujuan sebagai berikut.

1. Mampu mendeskripsikan keunggulan dan kelemahan yang dihadapi ketika melakukan pengujian terhadap aplikasi *mobile acc.one* secara manual dan otomatis (dengan *tool* uji Katalon).
2. Menunjukkan hasil pengujian terhadap fungsi-fungsi yang diuji pada aplikasi *mobile acc.one*.

1.5 Metode Penelitian

Adapun metode yang digunakan adalah sebagai berikut.

1. Menganalisis Alur Fungsi yang Akan Diuji

Tahap pertama merupakan menganalisis dan mempelajari alur fungsi yang akan diuji dengan cara menguji secara manual, juga dengan cara melihat

langkah-langkah yang didokumentasikan oleh perusahaan mengenai fungsionalitas tersebut.

2. Pembuatan Skenario *Test Case*

Di tahap ini membuat *test case* pengujian aplikasi *mobile acc.one* dengan Microsoft Excel, berdasarkan *use case* yang telah dibuat. *Test Case* yang dibuat mencakup skenario bersifat positif dan negatif. Skenario positif berarti akan menguji fungsionalitas dari aplikasi dengan hasil akhir *user* akan berhasil menggunakan aplikasi tersebut, di mana dalam skenario ini akan dimasukkan *value* yang sesuai dengan fungsinya. Sedangkan skenario negatif berarti akan menguji fungsionalitas aplikasi dengan hasil akhir *user* akan mendapatkan *error* atau tidak berhasil, dengan memberikan nilai yang tidak sesuai dengan yang diharapkan pada fungsi tertentu. Skenario positif akan ditunjukkan pada *basic path* sedangkan negatif akan ditunjukkan beberapa pada *alternate path*.

3. Melakukan *Functional Testing* Secara Manual

Tahap pengujian aplikasi secara manual merupakan tahap di mana skenario yang sebelumnya telah dibuat akan diimplementasikan pada pengujian aplikasi secara manual. Sistem akan diuji tanpa bantuan *Testing Software*. Pengujian ini dilakukan dengan memperhatikan parameter-parameter yang telah ditentukan, dan hasil akhir dari pengujian ini akan dibuat laporan sebagai pembandingan antara kedua metode sesuai dengan tujuan penelitian.

4. Melakukan *Functional Testing* Secara Otomatis

Tahap pengujian aplikasi secara terotomisasi merupakan tahap di mana skenario yang sebelumnya telah dibuat akan diimplementasikan pada *Testing Software* yaitu Katalon. Tahap ini berfungsi untuk mengevaluasi indikator-indikator sesuai dengan tujuan penelitian. Pengujian ini dilakukan dengan memperhatikan parameter-parameter yang telah ditentukan, dan hasil akhir dari pengujian ini akan dibuat laporan sebagai pembandingan antara kedua metode sesuai dengan tujuan penelitian.

5. Penulisan Laporan Akhir

Penyusunan dan penulisan laporan merupakan tahap terakhir dari penelitian dan pengembangan sistem ini. Penulisan laporan penelitian ini mencakup latar belakang, tinjauan pustaka, landasan teori, dan lain-lain. Langkah terakhir dari tahap ini yaitu menuliskan hasil dari penelitian ini beserta saran yang dapat membantu penelitian selanjutnya.

1.6 Sistematika Penulisan

Laporan tugas akhir ini disusun dengan sistematika sebagai berikut.

BAB I : Pendahuluan

Pada bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, alat dan bahan, metodologi penelitian, dan sistematika penulisan laporan yang terkait dengan proses penelitian ini.

BAB II : Tinjauan Pustaka

Bab ini berisi tentang penjelasan singkat dan ringkasan mengenai penelitian terdahulu yang berhubungan atau memiliki kesamaan dengan permasalahan yang akan dibahas oleh penulis di dalam tugas akhir ini.

BAB III : Landasan Teori

Pada bab ini berisi dasar-dasar teori yang melandasi serta mendukung dalam implementasi aplikasi beserta proses yang berhubungan dengan pembangunan aplikasi.

BAB IV : Analisis dan Implementasi Algoritma

Bab ini berisi penjelasan uraian analisis algoritma dan perancangan sistem yang akan dibuat.

BAB V : Hasil dan Pembahasan

Bab ini berisi penjelasan mengenai hasil dari penelitian ini yang berupa data-data hasil pengujian dan pembahasannya.

BAB VI : Kesimpulan dan Saran

Pada bab ini berisi kesimpulan mengenai penelitian yang telah dibuat beserta saran-saran yang berguna bagi pengembangan lebih lanjut.

BAB II. TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Beberapa penelitian menyangkut pengujian aplikasi telah dilakukan oleh beberapa pihak. Salah satu pihak yang meneliti adalah Hanifah, dkk mengenai penggunaan metode *blackbox* pada pengujian Sistem Informasi Surat Keluar Masuk (SIKEMAS). Penelitian ini memiliki tujuan untuk mengetahui kekurangan dari sistem supaya data yang dikeluarkan nantinya dapat sesuai dengan data *input* yang telah dieksekusi. Pada pengujian kali ini berfokus kepada *data input*, *interface*, penggunaan memori, serta kecepatan eksekusi data. Pengujian aplikasi ini menggunakan 10 jenis tipe pengujian antara lain *equivalence partitioning*, *comparison testing*, *sample testing*, *robustness testing*, *behavior testing*, *boundary value analysis*, *performance testing*, *requirement testing*, *endurance testing*, dan *cause-effect relationship testing* dengan hasil pengujian yang terbilang cukup baik, sehingga ada beberapa kesalahan dan kelemahan pada aplikasi yang dapat ditemukan. Pengujian ini juga dirasa dapat mempermudah *developer* untuk memperbaiki dan mengembangkan fungsionalitas yang ditemukan pada aplikasi sewaktu pengujian [7].

Penelitian yang dilakukan oleh Mursyidah dan Hidayat juga menguji menggunakan metode *blackbox testing* di mana mereka menguji sistem informasi akuntansi dana BOS atau Biaya Operasional Sekolah berbasis web. Fungsi dibuatnya web SIA BOS ini agar para pengelola dana BOS dapat melihat dan mengevaluasi setiap perilaku penggunaan dana. Sistem ini dirancang dan dibuat menggunakan metode *Rapid Application Development* (RAD) di mana model ini merupakan proses pengembangan *software* secara *linier sequential* yang berfokus pada siklus *development* yang singkat. Adapun *form* yang diuji dalam kasus ini yaitu login, menu edit tahun anggaran, serta menu tambah dan edit RKA. Berdasarkan hasil pengujian aplikasi SIA BOS dengan menggunakan metode *blackbox* tidak ditemukan masalah pada sistem. Dapat disimpulkan bahwa sistem informasi yang telah dibuat sesuai dengan spesifikasi perancangan di *use case diagram* [8].

Penelitian selanjutnya datang dari peneliti Suhar Janti yang membuat penelitian mengenai kebutuhan dan permasalahan pada sistem tes seleksi untuk penerimaan siswa baru pada SMP Hang Tuah 1 Jakarta. Peneliti mendapatkan solusi untuk merancang sebuah aplikasi animasi edukasi yang interaktif, di mana terdapat permainan tebak warna dalam aplikasi dengan tujuan mengetahui kemampuan konsentrasi pada siswa baru yang akan diseleksi. Dalam penelitian ini tidak banyak dibahas mengenai pengujian sistem, akan tetapi pengujian *blackbox* dan *whitebox* juga ikut ambil peran dalam tahapan pengembangan sistem model *waterfall*. Pengujian *whitebox* yang dilakukan berfokus terhadap jalannya logika pada permainan tebak warna, sedangkan pada pengujian *blackbox* peneliti berfokus pada *input* dan *output* pada tombol-tombol proses yang ada pada program tersebut. Berdasarkan pengujian *blackbox* dari 4 tombol pada program yaitu *button* pilihan jawaban, pengatur waktu dan pergantian soal, tampil nilai setelah pergantian jawaban yang dihasilkan, pengatur waktu dan pergantian soal, dan tombol menu, keluaran sesuai dengan yang diharapkan. Hasil dari rancangan aplikasi oleh peneliti terhadap program tebak warna ini adalah aplikasi ini telah layak digunakan berdasarkan respons dari 60 peserta di mana terdiri atas 50 siswa dan 10 guru untuk mengetahui kemampuan konsentrasi siswa baru nantinya [9].

Penelitian yang dilakukan oleh Alfaris, dkk yaitu pengujian *website* informasi pendaftaran santri dengan menggunakan metode *blackbox*. Adanya aplikasi ini didorong pada permasalahan yang terjadi di pondok pesantren Al Anwar II Semarang yang masih belum memanfaatkan teknologi dalam penyajian informasi data santri, juga keterlambatan yang dialami oleh penyaji karena *input* data masih dilakukan secara manual. Metode yang digunakan di penelitian ini antara lain, mengumpulkan data dengan observasi, studi literatur, dan wawancara. Tujuan dilakukannya pengujian sistem ini adalah untuk memastikan bahwa fungsionalitas berjalan sesuai dengan yang diharapkan. Pengujian dengan metode *blackbox* dilakukan pada fungsional *input*, *edit*, *update*, dan hapus data. Adapun hasil yang didapatkan dari implementasi *blackbox* adalah fungsi yang lancar sesuai dengan harapan [10].

Pengujian aplikasi Android *Action & Strategy* yang dilakukan oleh Cholifah, dkk juga menggunakan *blackbox* dalam implementasinya. Aplikasi yang dibuat untuk *sharing knowledge* bagi lembaga detasemen pelatihan menembak di Brimob POLRI, dibuat dengan teknologi *phonegap* di mana *platform* ini memungkinkan *developer* mengembangkan aplikasi web untuk membuat aplikasi dan disebar ke 7 *platform mobile* sekaligus antara lain iOS, Android, Symbian, Blackberry, Windows Phone, WebOS, dan Samsung Bada. Pengujian ini dilakukan secara kuantitatif dengan 2 tahap yaitu pengujian fungsionalitas aplikasi secara manual dengan bantuan sejumlah *user* dan pengujian kedua dilakukan oleh hanya 1 orang yaitu web admin. Pengujian juga dilakukan pada 2 *platform* yang berbeda, yaitu Android dan web. Fungsi-fungsi yang diuji yakni *sign in*, tentang kami, *close*, *profile*, *back*, *exit*, halaman konfirmasi, soal, soal lanjut otomatis, *exit* dari soal, materi, dan *exit* dari materi. Hasil yang didapatkan setelah pengujian fungsionalitas pada kedua *platform*, tidak ditemukan adanya kesalahan ataupun *bug* dalam sistem [11].

Penelitian dilakukan oleh Rahmadi, dkk untuk menguji aplikasi prediksi kelulusan SNMPTN dengan menggunakan metode *blackbox*. Pengujian ini menggunakan cara *Boundary Value Analysis* di mana *tester* menentukan nilai batas atas dan bawah berdasarkan data yang akan diuji. Dalam kasus ini peneliti melakukan pengujian pada fungsi tambah kelas pada aplikasi prediksi kelulusan SNMPTN. Dari penelitian tersebut didapatkan bahwa metode pengujian *blackbox* mudah digunakan karena hanya memerlukan batas atas dan bawah, juga masih terdapat beberapa fungsionalitas yang masih dapat menerima masukan yang tidak diharapkan sehingga keluaran yang dihasilkan juga kurang valid. Saran dari peneliti untuk aplikasi berdasarkan hasil uji dengan BVA, maka *input* data seharusnya dilengkapi dengan *exception* untuk menjamin bahwa data yang dimasukkan sesuai dengan yang diharapkan [12].

Penelitian mengenai pengujian aplikasi juga dilakukan oleh Jaya. Ia menguji aplikasi untuk *software* kantor digital di Politeknik Negeri Lampung. Metode yang diimplementasi pada penelitian ini adalah *blackbox testing*

dengan pendekatan *Boundary Value Analysis*. Pengujian ini bertujuan untuk mengetahui tingkat kesalahan yang terjadi pada *software*. Hasil yang didapatkan dari penelitian kali ini menunjukkan bahwa keberhasilan aplikasi mencapai 91,67%. Terdapat 3 *field* yang perlu diperbaiki demi meningkatkan kualitas dan kemampuan aplikasi dalam pengolahan data di kondisi normal maupun tidak [13].

Pengujian *software* juga dilakukan oleh Komarudin di mana ia menguji aplikasi sistem informasi SDIT Al Qudwa Punggur Lampung. Pengujian ini menggunakan metode *blackbox* dengan teknik *equivalence partitioning* untuk menguji akurasi dari aplikasi. Pada penelitian ini, digunakan 12 *form* dan setiap *form* dilakukan 5 kali pengujian, jadi jika ditotal terdapat 60 kali pengujian. Hasil dari pengujian ini terdapat 36 *form error* dan 24 *form* tidak ditemukan *error*. Kesalahan terjadi pada fungsi, struktur data, insialisasi, *performance*, dan tampilan. Peneliti menganjurkan bahwa sebaiknya pada aplikasi perlu menggunakan metode lainnya yang dapat dijadikan standar keamanan aset data juga informasi. Selain itu juga sebaiknya dilakukan pengujian keamanan secara intens dengan menggunakan metode *whitebox penetration testing* secara berkala oleh admin maupun pengelola aplikasi yang berwenang di sekolah tersebut [14].

Penelitian dilakukan oleh Irawan, dkk mengenai pengujian aplikasi Bank Sampah Karya Sentosa dengan mengamati set *use case* sebagai dasar dari pembuatan *test case*. Pengujian ini menggunakan metode *blackbox* dengan teknik *equivalence partitioning*. Dalam penelitian ini pengujian akan dilakukan dengan menggunakan 8 modul yang memiliki fungsional antara lain pengumpulan data anggota, jenis anggota, jenis sampah, jumlah tabungan, jenis barang, transaksi tabungan, transaksi barang, dan laporan. Hasil dari penelitian ini menunjukkan bahwa masih terdapat beberapa sistem validasi yang belum terpenuhi sesuai spesifikasi, meskipun telah diuji dari pengujian struktural [15].

Penelitian dilakukan oleh Suherman, dkk mengenai pengujian aplikasi sistem seleksi *sales* terbaik. Pengujian ini menggunakan metode *blackbox* dengan teknik *equivalence partitions*. Di dalam *form* yang akan diuji terdapat

field data dan kinerja *sales* selama di perusahaan. Jumlah dari fungsi yang diuji adalah 4 fungsi dengan fungsi *login* diuji 2 kali, fungsi *save* diuji 6 kali, fungsi *edit* diuji 1 kali, dan fungsi *delete* diuji 1 kali. Jika ditotal terdapat 10 pengujian dalam penelitian ini, dan terdapat 1 fungsi yang ditemukan *error* dan 3 fungsi sisanya tidak ditemukan *error*. Peneliti menyarankan untuk menggunakan teknik pengujian *blackbox* yang lainnya untuk dapat menemukan *error* yang lebih bervariasi seperti *boundary value* atau *error guessing*, sehingga kualitas aplikasi dapat dikembangkan jauh lebih baik dan dapat bermanfaat bagi *user* [16].

Penelitian dilakukan oleh Barus dan Siburian yang bertujuan untuk membandingkan *tool* pengujian otomatis yang diuji ke beberapa aplikasi Android yaitu Aplikasi Siantar City, Aplikasi Selendroid Test App, dan Employee Directory. Penelitian ini menerapkan 14 kriteria yang ditetapkan sebagai parameter yang baiknya ada dalam *tool* pengujian. Hasil dari penelitian ini didapatkan bahwa *tool* Selendroid banyak memenuhi kriteria dari pengujian tersebut, sedangkan UI Automator terbukti menjadi *tool* uji yang membutuhkan waktu paling cepat, serta *tool* Calabash adalah *tool* uji yang memakan memori paling sedikit. Sehingga, peneliti menyarankan dari ketiga *tools* uji tersebut, penguji dapat menggunakan UI Automator untuk menguji aplikasi Android, dan juga dikarenakan faktor instalasi dan eksekusi *test case* yang lebih mudah dibandingkan *tools* yang lainnya [17].

Penelitian mengenai pengujian aplikasi belajar tajwid dengan basis platform Android dilakukan oleh Herlinda, dkk. Pengujian ini dilakukan secara otomatis dengan *tool* uji yaitu Katalon. Herlinda, dkk menerapkan metode pengujian *blackbox*. Tujuan dari adanya pengujian ini yakni untuk melihat dan menganalisis keefektifan *automation testing tool* Katalon lebih unggul dibandingkan manual. Pengujian ini memakan waktu rata-rata sekitar 26,128 detik hingga 5 menit. Dari 121 langkah pengujian, terdapat 3 kegagalan di mana ada ketidaksesuaian pada ukuran layar dalam metode *landscape*, objek suara yang terus menyala apabila pindah ke halaman lain, dan adanya kegagalan eksekusi dikarenakan Katalon tidak dapat mendeteksi objek yang tidak masuk ke dalam tangkapan layar di *device view* [18].

Tabel 1. Perbandingan Penelitian

Pembanding	Komarudin [14]	Mursyidah dan Hidayat [8]	Jaya [13]	Suherman, dkk. [16]	Herlinda, dkk. [18]	*) Pratiwi
Platform	<i>Desktop</i>	<i>Website</i>	<i>Website</i>	<i>Desktop</i>	<i>Mobile</i>	<i>Mobile</i>
Metode	<i>Blackbox</i>	<i>Blackbox</i>	<i>Blackbox</i>	<i>Blackbox</i>	<i>Blackbox</i>	<i>Blackbox</i>
Masalah yang Diuji	Sistem informasi SDIT Al Qudwah Punggur Lampung.	Sistem informasi akuntansi Biaya Operasional Sekolah (BOS).	Sistem informasi Kantor Digital Politeknik Negeri Lampung.	Sistem informasi seleksi <i>sales</i> terbaik.	Aplikasi belajar tajwid.	Aplikasi <i>mobile</i> acc.one.
Cara Pengujian	Manual	Manual	Manual	Manual	Otomatis	Otomatis dan Manual.

Hasil Penelitian	Terdapat 36 <i>form error</i> dan 24 <i>form</i> tidak ditemukan <i>error</i> .	Tidak ditemukan <i>bug</i> pada penelitian ini.	Terdapat 3 <i>field</i> yang perlu diperbaiki. Keberhasilan sistem mencapai 91,67%.	Ditemukan 1 fungsi <i>error</i> , sedangkan 3 fungsi lainnya sudah sesuai.	Dari 121 langkah pengujian, ditemukan 3 kegagalan.	Ditemukan <i>bug</i> di 2 fungsi dari 10 fungsi yang diuji. Masih terdapat kelemahan dari pengujian otomatis dan manual.
------------------	---	---	---	--	--	--

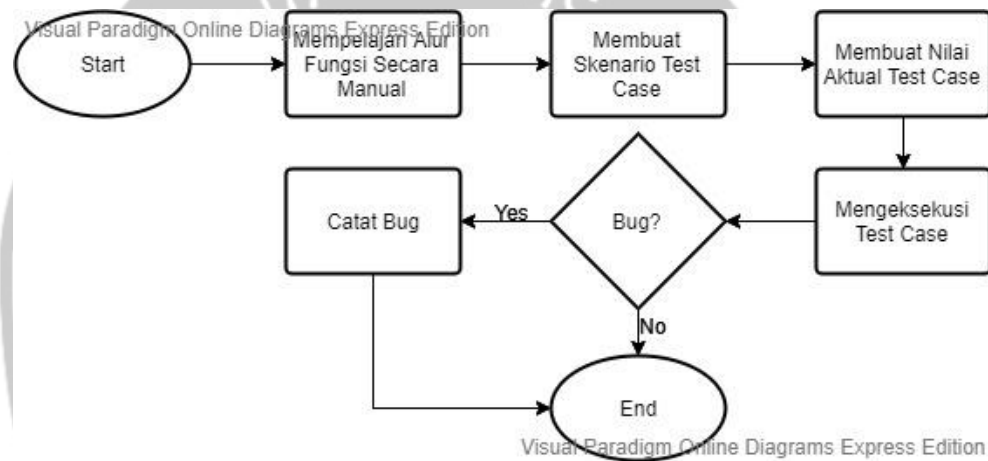
*) Penelitian yang dilakukan.

BAB V. HASIL EKSPERIMEN DAN PEMBAHASAN

5.1 Deskripsi Pengujian

5.1.1 Alur Pengujian Secara Manual

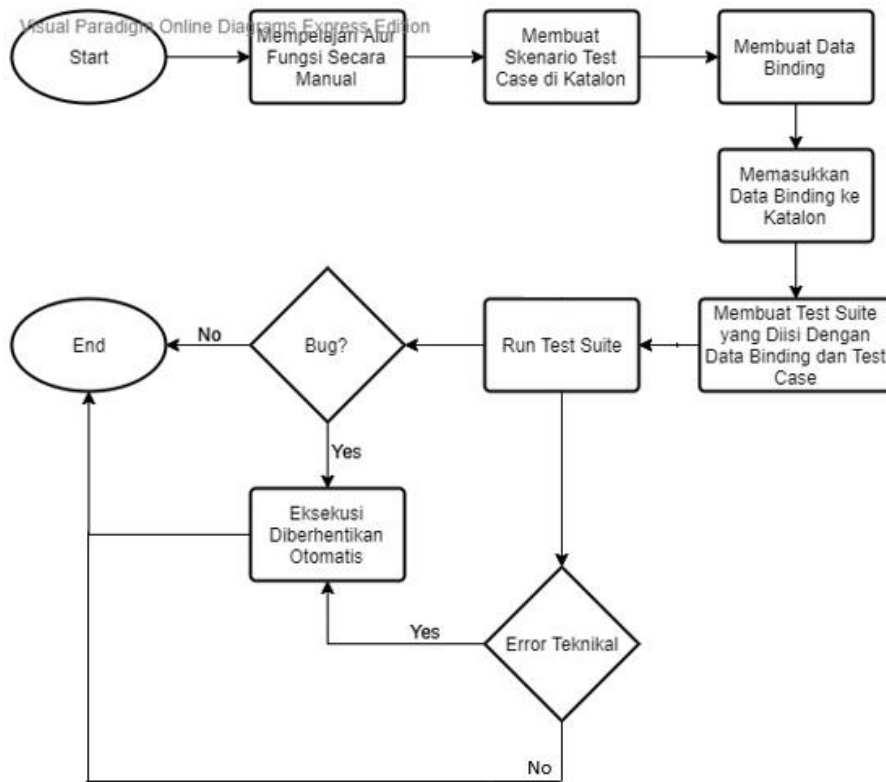
Pengujian secara manual dilakukan dengan merancang *test case* dan membuat *basic flow* secara manual, sehingga nantinya akan dicatat *output* yang dikeluarkan dari hasil pengujian dan dibandingkan dengan *expected output*, sehingga diperoleh nilai kesesuaian terhadap kedua parameter tersebut. Waktu juga akan dicatat pada pengujian manual dengan menggunakan *stopwatch*.



Gambar 16. Bagan Alur Pengujian acc.one Secara Manual.

5.1.2 Alur Pengujian Secara Otomatis

Pengujian secara otomatis dilakukan dengan cara merancang *test case* secara langsung di aplikasi Katalon dan membuat data *binding* atau nilai yang akan dimasukkan ke setiap variabel di Microsoft Excel. Kemudian data *binding* dan *test case* tersebut akan dimasukkan ke dalam *test suite* yang mana merupakan suatu media atau wadah yang telah disediakan Katalon untuk membantu *tester* mengeksekusi serangkaian *test case* yang telah dibuat dan melaporkan status dan hasil dari berhasil atau tidaknya pengujian. Waktu yang akan dihasilkan akan digenerate secara otomatis di Katalon.



Gambar 17. Bagan Alur Pengujian acc.one Secara Otomatis.

5.2 Hasil Pengujian

5.2.1 Analisis Hasil Pengujian Manual

5.2.1.1 Penerapan *Manual Testing* Untuk *Use Case* Mencari Mobil Baru

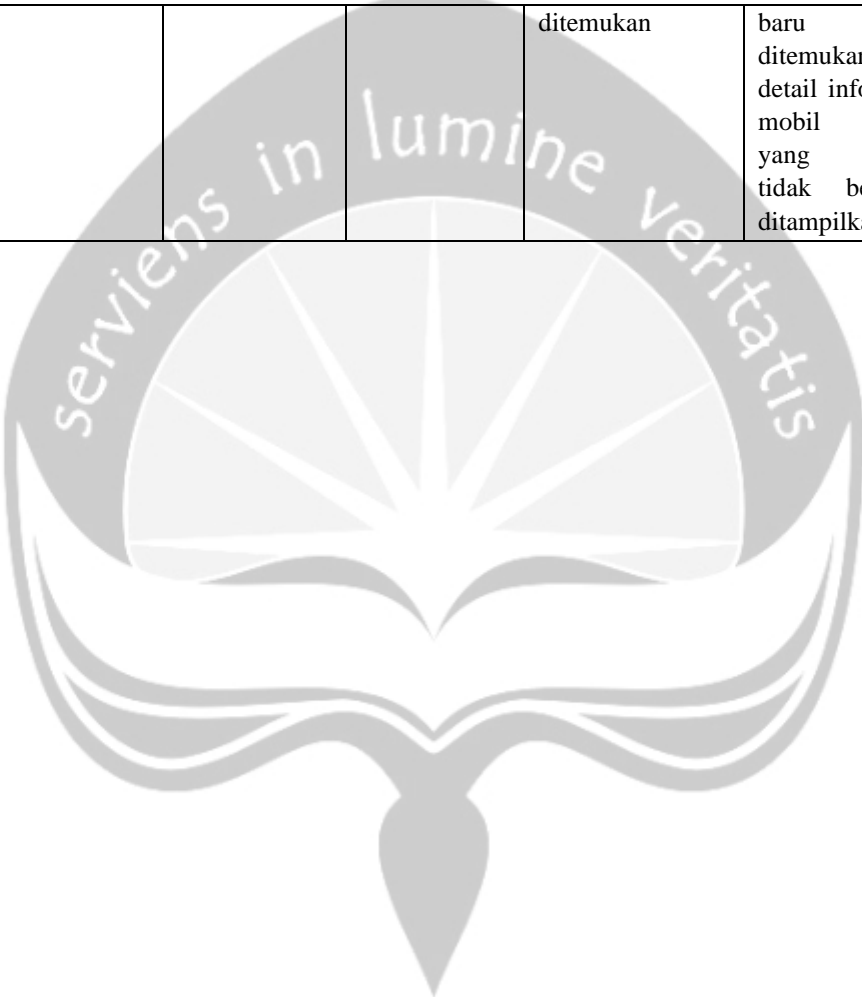
Berikut ini merupakan nilai aktual yang dirancang sesuai dengan matriks skenario yang telah dibuat di Bab 4. Nilai-nilai aktual dicantumkan pada Tabel 2 untuk *use case* mencari mobil baru. Nilai aktual diterapkan pada pengujian manual maupun otomatis. Pengujian dilakukan sebanyak 8 kali dengan 8 variasi skenario yang berbeda-beda dengan tujuan untuk menemukan kesalahan yang mungkin terjadi. Terdapat skenario positif dan negatif dalam pembuatan nilai aktual. Pengujian manual untuk *use case* ini dilakukan pada aplikasi acc.one dengan versi *update* tanggal 21 Januari 2020 dengan total waktu pengujian 5 menit 1 detik. Berdasarkan perbandingan *expected output* dan *output* yang dijabarkan pada Tabel 2, dapat ditarik kesimpulan bahwa pengujian *use case* mencari mobil baru secara manual memiliki persentase kesesuaian fungsionalitas 100%, yang berarti keluaran dari sistem semuanya sesuai dengan harapan awal.

Tabel 2. Nilai Aktual *Test Case* Untuk *Use Case* Mencari Mobil Baru.

<i>Test Case ID</i>	Skenario	namaMobil	spesific	brand	merek	<i>Expected Output</i>	<i>Output</i>	Keterangan	Hasil Akhir
TC1	Skenario 1: BF	Rush	Rush 1.5 G M/T	N/A	N/A	Informasi mobil baru ditampilkan	Detail informasi mobil baru yang dicari ditampilkan.	Pencarian mobil baru secara umum.	Sesuai.
TC2	Skenario 2: BF A1	Terios	All New Terios X A/T	Daihatsu	N/A	Informasi mobil baru ditampilkan	Detail informasi mobil baru yang dicari	Pencarian mobil baru berdasarkan <i>brand</i> .	Sesuai.

							ditampilkan.		
TC3	Skenario 2: BF A1	Agya	Toyota Agya 1.0	Toyota	N/A	Informasi mobil baru ditampilkan	Detail informasi mobil baru yang dicari ditampilkan.	Pencarian mobil baru berdasarkan <i>brand</i> .	Sesuai.
TC4	Skenario 3: BF A1 A3	N/A	All New Terios X A/T	Daihatsu	All New Terios	Informasi mobil baru ditampilkan	Detail informasi mobil baru yang dicari ditampilkan.	Pencarian mobil baru berdasarkan merek.	Sesuai.
TC5	Skenario 3: BF A1 A3	N/A	Toyota Agya 1.0	Toyota	Agya	Informasi mobil baru ditampilkan	Detail informasi mobil baru yang dicari ditampilkan.	Pencarian mobil baru berdasarkan merek.	Sesuai.
TC6	Skenario 4: BF A1 A4	Agya	N/A	Daihatsu	N/A	Informasi mobil baru tidak ditemukan	Detail informasi mobil baru yang dicari tidak berhasil ditampilkan.	Mobil baru berdasarkan <i>brand</i> tidak ditemukan.	Sesuai.
TC7	Skenario 4: BF A1 A4	Terios	N/A	Toyota	N/A	Informasi mobil baru tidak ditemukan	Detail informasi mobil baru yang dicari tidak berhasil ditampilkan.	Mobil baru berdasarkan <i>brand</i> tidak ditemukan.	Sesuai.
TC8	Skenario 5:	Brio	N/A	N/A	N/A	Informasi mobil baru tidak	Muncul notifikasi mobil	Mobil baru tidak ditemukan.	Sesuai.

	BF A2				ditemukan	baru tidak ditemukan dan detail informasi mobil baru yang dicari tidak berhasil ditampilkan.		
--	--------------	--	--	--	-----------	--	--	--



5.2.1.2 Penerapan *Manual Testing* Untuk *Use Case* Mencari Promo Mobil Tertentu

Berikut ini merupakan nilai aktual yang dirancang sesuai dengan matriks skenario yang telah dibuat di Bab 4. Nilai-nilai aktual dicantumkan pada Tabel 3 untuk *use case* mencari promo mobil tertentu. Nilai aktual diterapkan pada pengujian manual maupun otomatis. Pengujian dilakukan sebanyak 2 kali dengan 2 variasi skenario yang berbeda-beda dengan tujuan untuk menemukan kesalahan yang mungkin terjadi. Terdapat skenario positif dan negatif dalam pembuatan nilai aktual. Pengujian manual untuk *use case* ini dilakukan pada aplikasi acc.one dengan versi *update* tanggal 21 Januari 2020 dengan total waktu pengujian 1 menit 4 detik. Berdasarkan perbandingan *expected output* dan *output* yang dijabarkan pada Tabel 3, dapat ditarik kesimpulan bahwa pengujian *use case* mencari promo mencari mobil tertentu secara manual memiliki persentase kesesuaian fungsionalitas 100%, yang berarti keluaran dari sistem semuanya sesuai dengan harapan awal.

Tabel 3. Nilai Aktual *Test Case* Untuk *Use Case* Mencari Promo Mobil Tertentu.

<i>Test Case ID</i>	Skenario	namaPromo	<i>Expected Output</i>	<i>Output</i>	Keterangan	Hasil Akhir
TC1	Skenario 1: BF	Paket Hemat	Detail promo ditampilkan.	Detail informasi promo ditampilkan.	Menampilkan informasi detail promo dari halaman utama.	Sesuai.
TC2	Skenario 2: BF A1	<i>Hot Deals</i> Mazda	Maaf, promo tidak ditemukan.	Terdapat notifikasi bahwa promo tidak ditemukan dan detail informasi promo tidak berhasil ditampilkan.	Promo tidak tersedia.	Sesuai.

5.2.1.3 Penerapan *Manual Testing* Untuk *Use Case* Mencari Mobil Bekas (*Marketplace*)

Berikut ini merupakan nilai aktual yang dirancang sesuai dengan matriks skenario yang telah dibuat di Bab 4. Nilai-nilai aktual dicantumkan pada Tabel 4 untuk *use case* mencari mobil bekas (*marketplace*). Nilai aktual diterapkan pada pengujian manual maupun otomatis. Pengujian dilakukan sebanyak 8 kali dengan 8 variasi skenario yang berbeda-beda dengan tujuan untuk menemukan kesalahan yang mungkin terjadi. Terdapat skenario positif dan negatif dalam pembuatan nilai aktual. Pengujian manual untuk *use case* ini dilakukan pada aplikasi acc.one dengan versi *update* tanggal 21 Januari 2020 dengan total waktu pengujian 5 menit 17 detik.

Saat dilakukan pengujian secara manual, ditemukan *bug* ketika pencarian mobil pada *text field* pencarian mobil berdasarkan *brand*. Ketidaksihesuaian dapat didukung ketika pencarian dilakukan dengan *scroll* mobil secara biasa dan *tap* informasi mobil bekas yang diinginkan dan pencarian melalui pengetikan nama mobil pada *text field*, yang berarti terdapat kesalahan logika pada *code* pencarian mobil bekas berdasarkan *brand*. Berdasarkan perbandingan *expected output* dan *output* yang dijabarkan pada Tabel 4, dapat ditarik kesimpulan bahwa pengujian *use case* mencari mobil bekas (*marketplace*) secara manual memiliki persentase kesesuaian fungsionalitas 75%, yang berarti 2 keluaran dari sistem tidak sesuai dengan harapan awal dan 6 sisanya sesuai dengan harapan awal.

Tabel 4. Nilai Aktual *Test Case* Untuk *Use Case* Mencari Mobil Bekas (*Marketplace*).

<i>Test Case ID</i>	Skenario	namaMobil	spesific	brand	merek	<i>Expected Output</i>	<i>Output</i>	Keterangan	Hasil Akhir
TC1	Skenario 1: BF	Xenia	Daihatsu Xenia 1.3 R CSTM A/T Bensin	N/A	N/A	Informasi mobil bekas bekas marketplace ditampilkan	Detail informasi mobil bekas yang dicari ditampilkan.	Pencarian mobil bekas marketplace secara umum.	Sesuai.

TC2	Skenario 2: BF A1	Ayla	Daihatsu Ayla 1.2 R M/T	Daihatsu	N/A	Informasi mobil bekas <i>marketplace</i> ditampilkan	Mobil tidak ditemukan.	Pencarian mobil bekas <i>marketplace</i> berdasarkan <i>brand</i> .	<i>Bug</i> .
TC3	Skenario 2: BF A1	Calya	Toyota Calya 1.2 G M/T	Toyota	N/A	Informasi mobil bekas <i>marketplace</i> ditampilkan	Mobil tidak ditemukan.	Pencarian mobil bekas <i>marketplace</i> berdasarkan <i>brand</i> .	<i>Bug</i> .
TC4	Skenario 3: BF A1 A3	N/A	Daihatsu Ayla 1.2 R M/T	Daihatsu	New Ayla	Informasi mobil bekas <i>marketplace</i> ditampilkan.	Detail informasi mobil bekas yang dicari ditampilkan.	Pencarian mobil bekas <i>marketplace</i> berdasarkan merek.	Sesuai.
TC5	Skenario 3: BF A1 A3	N/A	Toyota Calya 1.2 G M/T	Toyota	Calya	Informasi mobil bekas <i>marketplace</i> ditampilkan.	Detail informasi mobil bekas yang dicari ditampilkan.	Pencarian mobil bekas <i>marketplace</i> berdasarkan merek.	Sesuai.
TC6	Skenario 4: BF A1 A4	Calya	N/A	Daihatsu	N/A	Informasi mobil bekas <i>marketplace</i> tidak ditemukan.	Muncul notifikasi mobil bekas tidak ditemukan dan detail informasi mobil bekas yang dicari tidak berhasil ditampilkan.	Mobil bekas <i>marketplace</i> berdasarkan <i>brand</i> tidak ditemukan.	Sesuai.

TC7	Skenario 4: BF A1 A4	Terios	N/A	Toyota	N/A	Informasi mobil bekas <i>marketplace</i> tidak ditemukan.	Muncul notifikasi mobil bekas tidak ditemukan dan detail informasi mobil bekas yang dicari tidak berhasil ditampilkan.	Mobil bekas <i>marketplace</i> berdasarkan <i>brand</i> tidak ditemukan.	Sesuai.
TC8	Skenario 5: BF A2	Brio	N/A	N/A	N/A	Informasi mobil bekas <i>marketplace</i> tidak ditemukan.	Tidak muncul notifikasi. Detail mobil bekas (<i>marketplace</i>) yang dicari tidak berhasil ditampilkan.	Mobil bekas <i>marketplace</i> tidak ditemukan.	Sesuai.

5.2.1.4 Penerapan *Manual Testing* Untuk *Use Case* Mencari Mobil Bekas (Lelang)

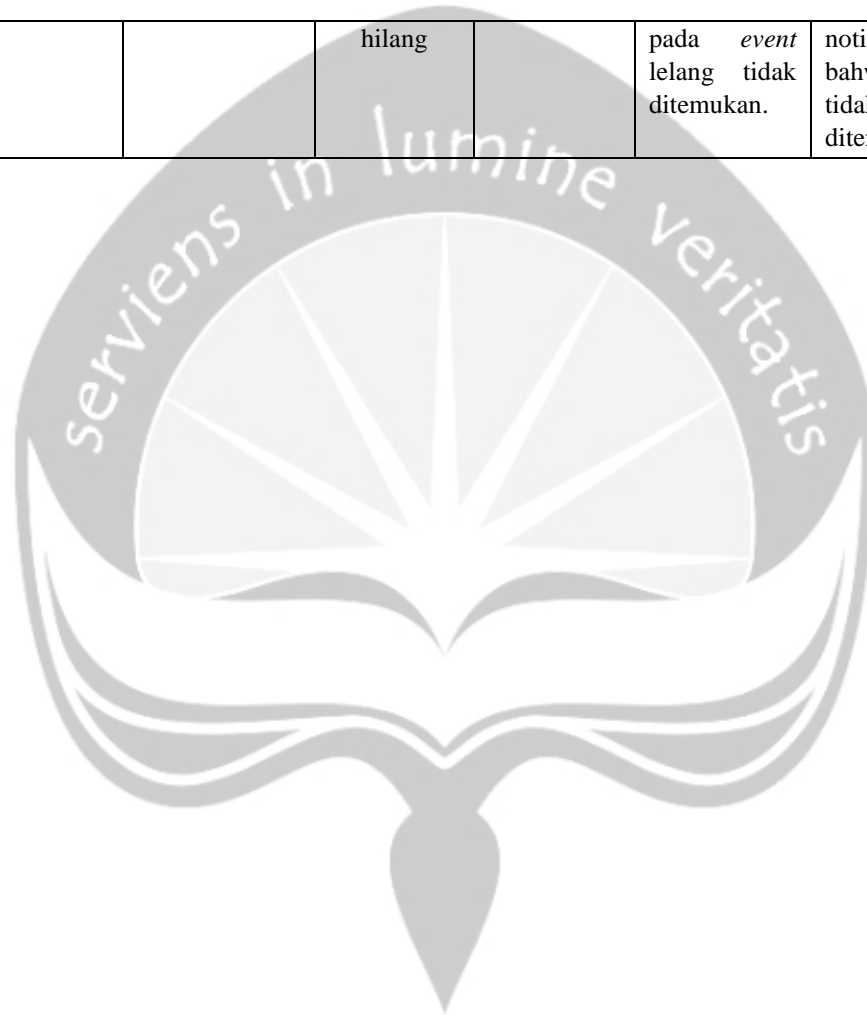
Berikut ini merupakan nilai aktual yang dirancang sesuai dengan matriks skenario yang telah dibuat di Bab 4. Nilai-nilai aktual dicantumkan pada Tabel 5 untuk *use case* mencari mobil bekas (lelang). Nilai aktual diterapkan pada pengujian manual maupun otomatis. Pengujian dilakukan sebanyak 8 kali dengan 8 variasi skenario yang berbeda-beda dengan tujuan untuk menemukan kesalahan yang mungkin terjadi. Terdapat skenario positif dan negatif dalam pembuatan nilai aktual. Pengujian manual untuk *use case* ini dilakukan pada aplikasi acc.one dengan versi *update* tanggal 21 Januari 2020 dengan total waktu pengujian 4 menit 48 detik. Berdasarkan perbandingan *expected output* dan *output* yang dijabarkan pada Tabel 5, dapat ditarik kesimpulan bahwa pengujian *use case* mencari mobil bekas (lelang) secara manual memiliki persentase kesesuaian fungsionalitas 100%, yang berarti keluaran dari sistem semuanya sesuai dengan harapan awal.

Tabel 5. Nilai Aktual *Test Case* Untuk *Use Case* Mencari Mobil Bekas (Lelang).

<i>Test Case ID</i>	<i>Skenario</i>	<i>Perusahaan Lelang</i>	<i>lokasi</i>	<i>tanggalLelang</i>	<i>namaEvent</i>	<i>namaMobil</i>	<i>Expected Output</i>	<i>Output</i>	<i>Keterangan</i>	<i>Hasil Akhir</i>
TC1	Skenario 1: BF	IBID	Batam	14 Maret 2020	jangan hilang	Chery QQ GX	Mobil bekas pada <i>event</i> lelang ditemukan.	Detail informasi mobil bekas lelang berhasil ditampilkan.	Menampilkan informasi detail mengenai mobil bekas pada <i>event</i> lelang yang dicari.	Sesuai.
TC2	Skenario 2: BF A1	IBID		N/A	N/A	N/A	Mobil bekas pada <i>event</i> lelang tidak ditemukan.	Terdapat peringatan lokasi harus dipilih.	Lokasi harus dipilih.	Sesuai.

TC3	Skenario 3: BF A2	IBID	Balikpapan	15 Maret 2020	N/A	N/A	Mobil bekas pada <i>event</i> lelang tidak ditemukan.	Terdapat notifikasi bahwa <i>event</i> lelang tidak tersedia pada tanggal tersebut.	Tidak terdapat <i>event</i> lelang pada tanggal tersebut.	Sesuai.
TC4	Skenario 4: BF A3	IBID	Batam	17 Maret 2020	jangan hilang	Agya	Mobil bekas pada <i>event</i> lelang tidak ditemukan.	Terdapat notifikasi bahwa mobil tidak ditemukan.	Mobil bekas tidak dapat ditemukan pada <i>event</i> lelang tersebut.	Sesuai.
TC5	Skenario 5: BF A1 A2	IBID	Balikpapan	15 Maret 2020	N/A	N/A	Mobil bekas pada <i>event</i> lelang ditemukan.	Terdapat notifikasi bahwa <i>event</i> lelang tidak tersedia pada tanggal tersebut.	Tidak terdapat <i>event</i> lelang pada tanggal tersebut.	Sesuai.
TC6	Skenario 6: BF A1 A3	IBID	Batam	17 Maret 2020	jangan hilang	Agya	Mobil bekas pada <i>event</i> lelang tidak ditemukan.	Terdapat notifikasi bahwa mobil tidak ditemukan.	Mobil bekas tidak dapat ditemukan pada <i>event</i> lelang tersebut.	Sesuai.
TC7	Skenario 7: BF A2 A3	IBID	Batam	17 Maret 2020	jangan hilang	Agya	Mobil bekas pada <i>event</i> lelang tidak ditemukan.	Terdapat notifikasi bahwa mobil tidak ditemukan.	Mobil bekas tidak dapat ditemukan pada <i>event</i> lelang tersebut.	Sesuai.
TC8	Skenario 8:	IBID	Batam	17 Maret 2020	jangan	Agya	Mobil bekas	Terdapat	Mobil bekas	Sesuai.

	BF A1 A2 A3				hilang	pada <i>event</i> lelang tidak ditemukan.	notifikasi bahwa mobil tidak ditemukan.	tidak dapat ditemukan pada <i>event</i> lelang tersebut.	
--	------------------------	--	--	--	--------	---	---	--	--



5.2.1.5 Penerapan *Manual Testing* Untuk *Use Case* Mengubah Foto Profil

Berikut ini merupakan nilai aktual yang dirancang sesuai dengan matriks skenario yang telah dibuat di Bab 4. Nilai-nilai aktual dicantumkan pada Tabel 6 untuk *use case* mengubah foto profil. Nilai aktual diterapkan pada pengujian manual maupun otomatis. Pengujian dilakukan sebanyak 6 kali dengan 6 variasi skenario yang berbeda-beda dengan tujuan untuk menemukan kesalahan yang mungkin terjadi. Terdapat skenario positif dan negatif dalam pembuatan nilai aktual. Pengujian manual untuk *use case* ini dilakukan pada aplikasi acc.one dengan versi *update* tanggal 21 Januari 2020 dengan total waktu pengujian 1 menit 44 detik. Berdasarkan perbandingan *expected output* dan *output* yang dijabarkan pada Tabel 6, dapat ditarik kesimpulan bahwa pengujian *use case* mengubah foto profil secara manual memiliki persentase kesesuaian fungsionalitas 100%, yang berarti keluaran dari sistem semuanya sesuai dengan harapan awal.

Tabel 6. Nilai Aktual *Test Case* Untuk *Use Case* Mengubah Foto Profil.

<i>Test Case ID</i>	Skenario	namaFoto	picture	<i>Expected Output</i>	<i>Output</i>	Keterangan	Hasil Akhir
TC1	Skenario 1: BF	gambar1.jpg	Galeri	Gambar berhasil diubah.	Muncul notifikasi gambar telah diubah dan gambar berhasil diubah.	Gambar berhasil <i>diupdate</i> . Gambar diambil dari galeri perangkat.	Sesuai.
TC2	Skenario 2: BF A1	gambar1.jpg	Kamera	Gambar berhasil diubah.	Muncul notifikasi gambar telah diubah dan gambar berhasil diubah.	Gambar berhasil <i>diupdate</i> . Gambar diambil dari galeri perangkat.	Sesuai.
TC3	Skenario 3: BF A1 A2	gambar1.jpg	Kamera	Gambar berhasil diubah.	Muncul notifikasi gambar telah diubah dan gambar berhasil diubah.	Gambar berhasil <i>diupdate</i> . Gambar diambil dari kamera perangkat. <i>Customer</i> memilih untuk mengambil ulang gambar.	Sesuai.

5.2.1.6 Penerapan *Manual Testing* Untuk *Use Case* Mencari Mobil Rental

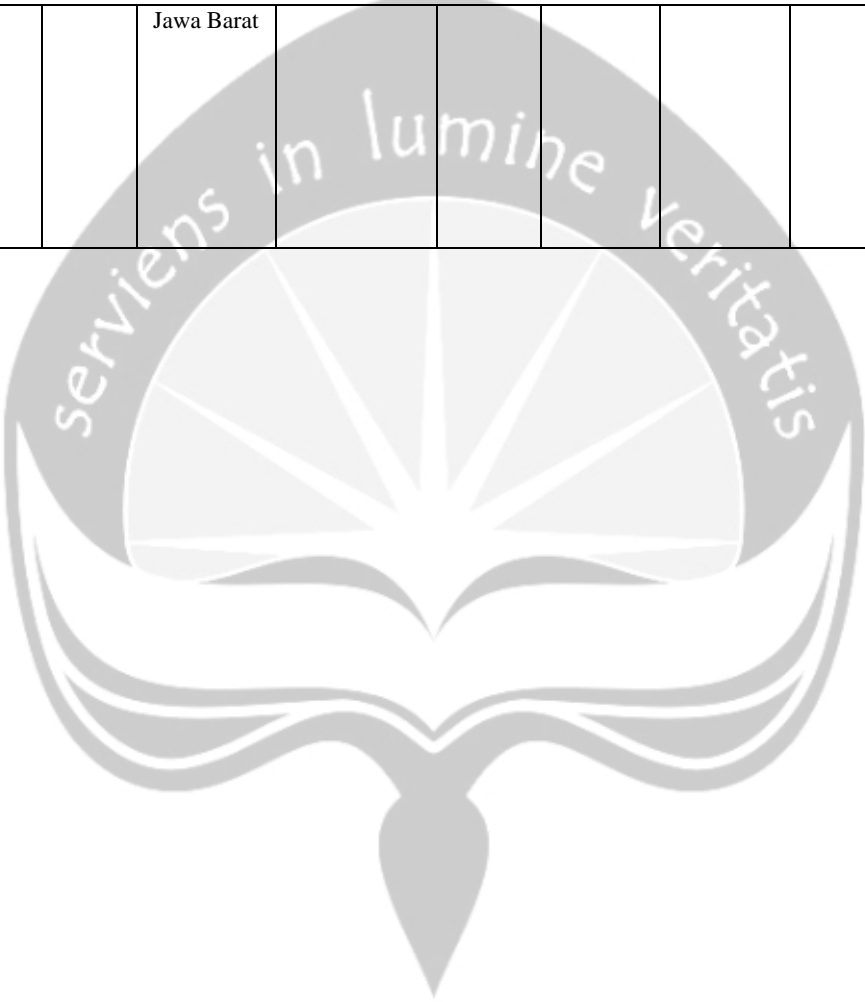
Berikut ini merupakan nilai aktual yang dirancang sesuai dengan matriks skenario yang telah dibuat di Bab 4. Nilai-nilai aktual dicantumkan pada Tabel 7 untuk *use case* mencari mobil rental. Nilai aktual diterapkan pada pengujian manual maupun otomatis. Pengujian dilakukan sebanyak 6 kali dengan 6 variasi skenario yang berbeda-beda dengan tujuan untuk menemukan kesalahan yang mungkin terjadi. Terdapat skenario positif dan negatif dalam pembuatan nilai aktual. Pengujian manual untuk *use case* ini dilakukan pada aplikasi acc.one dengan versi *update* tanggal 21 Januari 2020 dengan total waktu pengujian 5 menit 52 detik. Berdasarkan perbandingan *expected output* dan *output* yang dijabarkan pada Tabel 7, dapat ditarik kesimpulan bahwa pengujian *use case* mencari mobil rental secara manual memiliki persentase kesesuaian fungsionalitas 100%, yang berarti keluaran dari sistem semuanya sesuai dengan harapan awal.

Tabel 7. Nilai Aktual *Test Case* Untuk *Use Case* Mencari Mobil Rental.

<i>Test Case ID</i>	Skenario	newOr Used	brand	typeMobil	model	tahun	area Pengajuan	cabangACC	tujuan	bidang Usaha	type Customer	Warna	periode Leasing	<i>Expected Output</i>	<i>Output</i>	Keterangan	Hasil Akhir
TC1	Skenario 1: BF	new	TOYOTA	ALL NEW AVANZA	1.3 E A/T	2019	DKI Jakarta, Banten, Jawa Barat	KARAWACI	Pribadi	N/A	Personal	Hitam, Putih	48	Mobil sedang diproses.	Muncul notifikasi mobil sedang diproses.	Mobil berhasil diproses.	Sesuai.
TC2	Skenario 1: BF A1	used	DAIHATSU	AYLA	T:1.0 D+M/T NEW	2018	DKI Jakarta, Banten, Jawa Barat	PLUIT	Usaha	Pertanian	Company	Silver, Merah	36	Mobil sedang diproses.	Muncul notifikasi mobil sedang diproses.	Tujuan penggunaan untuk Usaha.	Sesuai.
TC3	Skenario 1: BF A1 A3	used	DAIHATSU	AYLA	T:1.0 D+M/T NEW	2018	DKI Jakarta, Banten,	PLUIT	Usaha		Company	Putih, Silver, Merah	36	Muncul peringatan pada salah	Muncul notifikasi harus	Tujuan penggunaan untuk	Sesuai.

							Jawa Barat							satu variabel bahwa harus dipilih.	diisi pada Bidang Usaha.	Usaha. <i>Dropdown</i> Bidang Usaha harus dipilih.	
TC4	Skenario 1: BF A1 A3 A2	<i>used</i>	DAIHATSU	AYLA	T:1.0 D+M/T NEW	2018	DKI Jakarta, Banten, Jawa Barat		Usaha		Company	Semua Warna	24	Muncul peringatan pada salah satu variabel bahwa harus dipilih.	Muncul notifikasi harus diisi pada Bidang Usaha dan pada salah satu variabel yang belum dipilih.	Tujuan penggunaan untuk Usaha. <i>Dropdown</i> Bidang Usaha harus dipilih. Terdapat 1 atau beberapa variabel yang belum dan harus dipilih.	Sesuai.
TC5	Skenario 1: BF A1 A2	<i>used</i>	DAIHATSU	AYLA	T:1.0 D+M/T NEW	2018	DKI Jakarta, Banten, Jawa Barat		Usaha	Pertanian	Company	Semua Warna	24	Muncul peringatan pada salah satu variabel bahwa harus dipilih.	Muncul notifikasi harus diisi pada salah satu variabel yang belum dipilih.	Tujuan penggunaan untuk Usaha. Terdapat 1 atau beberapa variabel yang belum dan harus dipilih.	Sesuai.
TC6	Skenario 1: BF A2	<i>new</i>	TOYOTA	ALL NEW AVANZA	1.3 E A/T	2019	DKI Jakarta, Banten,		Pribadi	N/A	Personal	Putih, Silver	60	Muncul peringatan pada salah	Muncul notifikasi harus	Terdapat 1 atau beberapa	Sesuai.

							Jawa Barat											satu variabel bahwa harus dipilih.	diisi pada salah satu variabel yang belum dipilih.	variabel yang belum dan harus dipilih.	
--	--	--	--	--	--	--	------------	--	--	--	--	--	--	--	--	--	--	------------------------------------	--	--	--



5.2.1.7 Penerapan *Manual Testing* Untuk *Use Case* Lupa Kata Sandi

Berikut ini merupakan nilai aktual yang dirancang sesuai dengan matriks skenario yang telah dibuat di Bab 4. Nilai-nilai aktual dicantumkan pada Tabel 8 untuk *use case* lupa kata sandi. Nilai aktual diterapkan pada pengujian manual maupun otomatis. Pengujian dilakukan sebanyak 9 kali dengan 9 variasi skenario yang berbeda-beda dengan tujuan untuk menemukan kesalahan yang mungkin terjadi. Skenario yang dijalankan antara lain: TC1, TC2, TC3, TC4, TC5, TC6, TC7, TC8, dan TC9, dikarenakan 9 skenario tersebut sudah mewakili semua *alternate flow* yang telah dirancang di Bab 4. Terdapat skenario positif dan negatif dalam pembuatan nilai aktual. Pengujian manual untuk *use case* ini dilakukan pada aplikasi acc.one dengan versi *update* tanggal 12 Februari 2020 dengan total waktu pengujian 10 menit 8 detik.

Berdasarkan analisis alur secara manual, sistem memberikan 2 notifikasi yang muncul pada saat sebelum dan sesudah *tap button* reset dan masuk, dan kedua notifikasi akan menampilkan teks yang berbeda ketika sebelum dan sesudah *tap. Button* reset dan masuk juga akan bersifat *not visible* apabila kata sandi baru dan konfirmasi belum pernah diisi. Jadi, ketika nilai aktual diisi kosong seperti yang tertera di Tabel 8, pada implementasinya diisi dengan spasi atau 1 karakter, kemudian dihapus untuk memunculkan *button* reset dan masuk pada saat pengujian manual. Hasil yang didapatkan dari pengujian adalah terdapat 1 *bug* di mana jika memasukkan 7 karakter alfanumerik pada *text field* kata sandi baru, notifikasi ‘minimal 7 karakter’ masih akan tetap muncul padahal terdapat catatan minimal 7 karakter alfanumerik pada saat memasukkan kata sandi. Sedangkan apabila masukan lebih dari 7 karakter alfanumerik, maka notifikasi tidak akan muncul. Ketidaksesuaian pada fungsi ini dipertegas ketika masukan akan tetap bernilai valid dan notifikasi hilang saat *tap button* reset dan masuk, yang berarti terdapat *logic error* pada *exception* di *text field* kata sandi baru. Jika ditarik kesimpulan dari keseluruhan perbandingan *expected output* dan *output* yang dijabarkan pada Tabel 8, pengujian *use case* lupa kata sandi secara manual memiliki persentase kesesuaian fungsionalitas 88,89%, yang berarti 8 dari 9 keluaran dari sistem sesuai dengan harapan awal.

Tabel 8. Nilai Aktual *Test Case* Untuk *Use Case* Lupa Kata Sandi.

<i>Test Case ID</i>	<i>Skenario</i>	<i>email</i>	<i> kirimOTPke</i>	<i>OTP</i>	<i>kataSandi</i>	<i>kataSandi2</i>	<i>Expected Output</i>	<i>Output</i>	<i>Keterangan</i>	<i>Hasil Akhir</i>
TC1	Skenario 1 : BF	rizkariz20@gmail.com	email	261294	A123456	A123456	Kata sandi berhasil diubah.	Muncul notifikasi kata sandi minimal 7 karakter, dan kata sandi berhasil diubah.	Kata sandi berhasil diubah.	<i>Bug.</i>
TC2	Skenario 2 : BF A1	tralalala@gmail.com	N/A	N/A	N/A	N/A	Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi email tidak terdaftar.	Email tidak terdaftar.	Sesuai.
TC3	Skenario 3 : BF A1 A3	rizkariz20@gmail.com	email	444444	N/A	N/A	Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi kode OTP salah.	OTP salah.	Sesuai.
TC4	Skenario 4 : BF A1 A3 A4	rizkariz20@gmail.com	email	261294			Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi kata sandi harus diisi dan kata sandi minimal 7 karakter.	Kata sandi baru kosong.	Sesuai.

TC5	Skenario 4 : BF A1 A3 A4	rizkariz20@gmail.com	email	261294	IN456		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi kata sandi minimal 7 karakter.	Kata sandi baru minimal 7 karakter.	Sesuai.
TC6	Skenario 4 : BF A1 A3 A4	rizkariz20@gmail.com	email	261294	12345678		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah. Muncul notifikasi format kata sandi tidak sesuai dan kata sandi minimal terdapat 1 huruf kecil.	Kata sandi baru harus minimal terdapat 1 huruf kecil atau besar.	Sesuai.
TC7	Skenario 4 : BF A1 A3 A4	rizkariz20@gmail.com	email	261294	AAAAAAAA		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah. Muncul notifikasi format kata sandi tidak sesuai dan kata sandi minimal terdapat 1 angka.	Kata sandi baru harus minimal terdapat 1 angka.	Sesuai.
TC8	Skenario 5 : BF A1 A3 A4 A5	rizkariz20@gmail.com	email	261294	A123456	IN1234	Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata sandi belum sesuai.	Konfirmasi kata sandi belum sesuai dengan kata sandi baru.	Sesuai.

TC9	Skenario 5 : BF A1 A3 A4 A5	rizkariz20@gmail.com	email	261294	A123456		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata sandi harus diisi.	Konfirmasi kata sandi masih kosong.	Sesuai.
TC10	Skenario 6 : BF A1 A3 A5	rizkariz20@gmail.com	email	261294	A123456	IN1234	Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata sandi belum sesuai.	Konfirmasi kata sandi belum sesuai dengan kata sandi baru.	Sesuai.
TC11	Skenario 6 : BF A1 A3 A5	rizkariz20@gmail.com	email	261294	A123456		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata sandi harus diisi.	Konfirmasi kata sandi masih kosong.	Sesuai.
TC12	Skenario 7 : BF A1 A4	rizkariz20@gmail.com	email	261294			Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi kata sandi baru harus diisi.	Kata sandi baru kosong.	Sesuai.
TC13	Skenario 7 : BF A1 A4	rizkariz20@gmail.com	email	261294	IN456		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi kata sandi minimal 7	Kata sandi baru minimal 7 karakter.	Sesuai.

								karakter.		
TC14	Skenario 7 : BF A1 A4	rizkariz20@gmail.com	email	261294	12345678		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah. Muncul notifikasi format kata sandi tidak sesuai dan kata sandi minimal terdapat 1 huruf kecil.	Kata sandi baru harus minimal terdapat 1 huruf kecil atau besar.	Sesuai.
TC15	Skenario 7 : BF A1 A4	rizkariz20@gmail.com	email	261294	AAAAAAAA		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah. Muncul notifikasi format kata sandi tidak sesuai dan kata sandi minimal terdapat 1 angka.	Kata sandi baru harus minimal terdapat 1 angka.	Sesuai.
TC16	Skenario 8 : BF A1 A4 A5	rizkariz20@gmail.com	email	261294	A123456	IN1234	Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata sandi belum sesuai.	Konfirmasi kata sandi belum sesuai dengan kata sandi baru.	Sesuai.
TC17	Skenario 8 : BF A1 A4 A5	rizkariz20@gmail.com	email	261294	A123456		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata	Konfirmasi kata sandi masih kosong.	Sesuai.

								sandi harus diisi.		
TC18	Skenario 9 : BF A1 A5	rizkariz20@gmail.com	email	261294	A123456	IN1234	Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata sandi belum sesuai.	Konfirmasi kata sandi belum sesuai dengan kata sandi baru.	Sesuai.
TC19	Skenario 9 : BF A1 A5	rizkariz20@gmail.com	email	261294	A123456		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata sandi harus diisi.	Konfirmasi kata sandi masih kosong.	Sesuai.
TC20	Skenario 10 : BF A3	rizkariz20@gmail.com	email	444444	N/A	N/A	Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi kode OTP salah.	OTP salah.	Sesuai.
TC21	Skenario 11 : BF A3 A4	rizkariz20@gmail.com	email	261294	INA123456		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi kata sandi baru harus diisi.	Kata sandi baru kosong.	Sesuai.
TC22	Skenario 11 : BF A3 A4	rizkariz20@gmail.com	email	261294	INA123456		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi kata	Kata sandi baru minimal 7 karakter.	Sesuai.

								sandi minimal 7 karakter.		
TC23	Skenario 11 : BF A3 A4	rizkariz20@gmail.com	email	261294	12345678		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah. Muncul notifikasi format kata sandi tidak sesuai dan kata sandi minimal terdapat 1 huruf kecil.	Kata sandi baru harus minimal terdapat 1 huruf kecil atau besar.	Sesuai.
TC24	Skenario 11 : BF A3 A4	rizkariz20@gmail.com	email	261294	AAAAAAAA		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah. Muncul notifikasi format kata sandi tidak sesuai dan kata sandi minimal terdapat 1 angka.	Kata sandi baru harus minimal terdapat 1 angka.	Sesuai.
TC25	Skenario 12 : BF A3 A4 A5	rizkariz20@gmail.com	email	261294	A123456	IN1234	Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata sandi belum sesuai.	Konfirmasi kata sandi belum sesuai dengan kata sandi baru.	Sesuai.
TC26	Skenario 12 : BF A3 A4 A5	rizkariz20@gmail.com	email	261294	A123456		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi	Konfirmasi kata sandi masih kosong.	Sesuai.

								konfirmasi kata sandi harus diisi.		
TC27	Skenario 13 : BF A3 A5	rizkariz20@gmail.com	email	261294	A123456	IN1234	Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata sandi belum sesuai.	Konfirmasi kata sandi belum sesuai dengan kata sandi baru.	Sesuai.
TC28	Skenario 13 : BF A3 A5	rizkariz20@gmail.com	email	261294	A123456		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata sandi harus diisi.	Konfirmasi kata sandi masih kosong.	Sesuai.
TC29	Skenario 14 : BF A4	rizkariz20@gmail.com	email	261294			Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi kata sandi baru harus diisi.	Kata sandi baru kosong.	Sesuai.
TC30	Skenario 14 : BF A4	rizkariz20@gmail.com	email	261294	IN456		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi kata sandi minimal 7 karakter.	Kata sandi baru minimal 7 karakter.	Sesuai.
TC31	Skenario 14 : BF A4	rizkariz20@gmail.com	email	261294	12345678		Kata sandi tidak berhasil	Kata sandi tidak berhasil diubah. Muncul	Kata sandi baru harus minimal terdapat 1	Sesuai.

							diubah.	notifikasi format kata sandi tidak sesuai dan kata sandi minimal terdapat 1 huruf kecil.	huruf kecil atau besar.	
TC32	Skenario 14 : BF A4	rizkariz20@gmail.com	email	261294	AAAAAAAA		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah. Muncul notifikasi format kata sandi tidak sesuai dan kata sandi minimal terdapat 1 angka.	Kata sandi baru harus minimal terdapat 1 angka.	Sesuai.
TC33	Skenario 15 : BF A4 A5	rizkariz20@gmail.com	email	261294	A123456	IN1234	Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata sandi belum sesuai.	Konfirmasi kata sandi belum sesuai dengan kata sandi baru.	Sesuai.
TC34	Skenario 15 : BF A4 A5	rizkariz20@gmail.com	email	261294	A123456		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata sandi harus diisi.	Konfirmasi kata sandi masih kosong.	Sesuai.
TC35	Skenario 16 : BF A5	rizkariz20@gmail.com	email	261294	A123456	IN1234	Kata sandi tidak berhasil	Kata sandi tidak berhasil diubah dan muncul	Konfirmasi kata sandi belum sesuai	Sesuai.

							diubah.	notifikasi konfirmasi kata sandi belum sesuai.	dengan kata sandi baru.	
TC36	Skenario 16 : BF A5	rizkariz20@gmail.com	email	261294	A123456		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata sandi harus diisi.	Konfirmasi kata sandi masih kosong.	Sesuai.
TC37	Skenario 17 : BF A1 A2	rizkariz20@gmail.com	email	261294	A123456	A123456	Kata sandi berhasil diubah.	Muncul notifikasi kata sandi minimal 7 karakter, dan kata sandi berhasil diubah.	<i>Tap button</i> submit Captcha untuk verifikasi.	<i>Bug.</i>
TC38	Skenario 18 : BF A1 A2 A3	rizkariz20@gmail.com	email	444444	N/A	N/A	Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi kode OTP salah.	OTP salah.	Sesuai.
TC39	Skenario 19 : BF A1 A2 A3 A4	rizkariz20@gmail.com	email	261294			Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi kata sandi harus diisi dan kata sandi minimal 7 karakter.	Kata sandi baru kosong.	Sesuai.

TC40	Skenario 19 : BF A1 A2 A3 A4	rizkariz20@gmail.com	email	261294	IN456		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi kata sandi minimal 7 karakter.	Kata sandi baru minimal 7 karakter.	Sesuai.
TC41	Skenario 19 : BF A1 A2 A3 A4	rizkariz20@gmail.com	email	261294	12345678		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah. Muncul notifikasi format kata sandi tidak sesuai dan kata sandi minimal terdapat 1 huruf kecil.	Kata sandi baru harus minimal terdapat 1 huruf kecil atau besar.	Sesuai.
TC42	Skenario 19 : BF A1 A2 A3 A4	rizkariz20@gmail.com	email	261294	AAAAAAAA		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah. Muncul notifikasi format kata sandi tidak sesuai dan kata sandi minimal terdapat 1 angka.	Kata sandi baru harus minimal terdapat 1 angka.	Sesuai.
TC43	Skenario 20 : BF A1 A2 A3 A4 A5	rizkariz20@gmail.com	email	261294	A123456	IN1234	Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata sandi belum sesuai.	Konfirmasi kata sandi belum sesuai dengan kata sandi baru.	Sesuai.

TC44	Skenario 20 : BF A1 A2 A3 A4 A5	rizkariz20@gmail.com	email	261294	A123456		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata sandi harus diisi.	Konfirmasi kata sandi masih kosong.	Sesuai.
TC45	Skenario 21 : BF A1 A2 A3 A5	rizkariz20@gmail.com	email	261294	A123456	IN1234	Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata sandi belum sesuai.	Konfirmasi kata sandi belum sesuai dengan kata sandi baru.	Sesuai.
TC46	Skenario 21 : BF A1 A2 A3 A5	rizkariz20@gmail.com	email	261294	A123456		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata sandi harus diisi.	Konfirmasi kata sandi masih kosong.	Sesuai.
TC47	Skenario 22 : BF A1 A2 A4	rizkariz20@gmail.com	email	261294			Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi kata sandi baru harus diisi.	Kata sandi baru kosong.	Sesuai.
TC48	Skenario 22 : BF A1 A2 A4	rizkariz20@gmail.com	email	261294	IN456		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi kata sandi minimal 7	Kata sandi baru minimal 7 karakter.	Sesuai.

								karakter.		
TC49	Skenario 22 : BF A1 A2 A4	rizkariz20@gmail.com	email	261294	12345678		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah. Muncul notifikasi format kata sandi tidak sesuai dan kata sandi minimal terdapat 1 huruf kecil.	Kata sandi baru harus minimal terdapat 1 huruf kecil atau besar.	Sesuai.
TC50	Skenario 22 : BF A1 A2 A4	rizkariz20@gmail.com	email	261294	AAAAAAAA		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah. Muncul notifikasi format kata sandi tidak sesuai dan kata sandi minimal terdapat 1 angka.	Kata sandi baru harus minimal terdapat 1 angka.	Sesuai.
TC51	Skenario 23 : BF A1 A2 A4 A5	rizkariz20@gmail.com	email	261294	A123456	IN1234	Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata sandi belum sesuai.	Konfirmasi kata sandi belum sesuai dengan kata sandi baru.	Sesuai.
TC52	Skenario 23 : BF A1 A2 A4 A5	rizkariz20@gmail.com	email	261294	A123456		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata	Konfirmasi kata sandi masih kosong.	Sesuai.

								sandi harus diisi.		
TC53	Skenario 24 : BF A1 A2 A5	rizkariz20@gmail.com	email	261294	A123456	IN1234	Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata sandi belum sesuai.	Konfirmasi kata sandi belum sesuai dengan kata sandi baru.	Sesuai.
TC54	Skenario 24 : BF A1 A2 A5	rizkariz20@gmail.com	email	261294	A123456		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata sandi harus diisi.	Konfirmasi kata sandi masih kosong.	Sesuai.
TC55	Skenario 25 : BF A2	rizkariz20@gmail.com	email	261294	A123456	A123456	Kata sandi berhasil diubah.	Muncul notifikasi kata sandi minimal 7 karakter, dan kata sandi berhasil diubah.	<i>Tap button</i> submit Captcha untuk verifikasi.	<i>Bug.</i>
TC56	Skenario 26 : BF A2 A3	rizkariz20@gmail.com	email	444444	N/A	N/A	Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi kode OTP salah.	OTP salah.	Sesuai.
TC57	Skenario 27 : BF A2 A3 A4	rizkariz20@gmail.com	email	261294	INA123456		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi kata	Kata sandi baru kosong.	Sesuai.

								sandi baru harus diisi.		
TC58	Skenario 27 : BF A2 A3 A4	rizkariz20@gmail.com	email	261294	INA123456		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi kata sandi minimal 7 karakter.	Kata sandi baru minimal 7 karakter.	Sesuai.
TC59	Skenario 27 : BF A2 A3 A4	rizkariz20@gmail.com	email	261294	12345678		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah. Muncul notifikasi format kata sandi tidak sesuai dan kata sandi minimal terdapat 1 huruf kecil.	Kata sandi baru harus minimal terdapat 1 huruf kecil atau besar.	Sesuai.
TC60	Skenario 27 : BF A2 A3 A4	rizkariz20@gmail.com	email	261294	AAAAAAAA		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah. Muncul notifikasi format kata sandi tidak sesuai dan kata sandi minimal terdapat 1 angka.	Kata sandi baru harus minimal terdapat 1 angka.	Sesuai.
TC61	Skenario 28 : BF A2 A3 A4 A5	rizkariz20@gmail.com	email	261294	A123456	IN1234	Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata	Konfirmasi kata sandi belum sesuai dengan kata sandi baru.	Sesuai.

								sandi belum sesuai.		
TC62	Skenario 28 : BF A2 A3 A4 A5	rizkariz20@gmail.com	email	261294	A123456		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata sandi harus diisi.	Konfirmasi kata sandi masih kosong.	Sesuai.
TC63	Skenario 29 : BF A2 A3 A5	rizkariz20@gmail.com	email	261294	A123456	IN1234	Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata sandi belum sesuai.	Konfirmasi kata sandi belum sesuai dengan kata sandi baru.	Sesuai.
TC64	Skenario 29 : BF A2 A3 A5	rizkariz20@gmail.com	email	261294	A123456		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata sandi harus diisi.	Konfirmasi kata sandi masih kosong.	Sesuai.
TC65	Skenario 30 : BF A2 A4	rizkariz20@gmail.com	email	261294			Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi kata sandi baru harus diisi.	Kata sandi baru kosong.	Sesuai.
TC66	Skenario 30 : BF A2 A4	rizkariz20@gmail.com	email	261294	IN456		Kata sandi tidak berhasil	Kata sandi tidak berhasil diubah dan muncul	Kata sandi baru minimal 7 karakter.	Sesuai.

							diubah.	notifikasi kata sandi minimal 7 karakter.		
TC67	Skenario 30 : BF A2 A4	rizkariz20@gmail.com	email	261294	12345678		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah. Muncul notifikasi format kata sandi tidak sesuai dan kata sandi minimal terdapat 1 huruf kecil.	Kata sandi baru harus minimal terdapat 1 huruf kecil atau besar.	Sesuai.
TC68	Skenario 30 : BF A2 A4	rizkariz20@gmail.com	email	261294	AAAAAAAA		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah. Muncul notifikasi format kata sandi tidak sesuai dan kata sandi minimal terdapat 1 angka.	Kata sandi baru harus minimal terdapat 1 angka.	Sesuai.
TC69	Skenario 31 : BF A2 A4 A5	rizkariz20@gmail.com	email	261294	A123456	IN1234	Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata sandi belum sesuai.	Konfirmasi kata sandi belum sesuai dengan kata sandi baru.	Sesuai.
TC70	Skenario 31 : BF A2 A4 A5	rizkariz20@gmail.com	email	261294	A123456		Kata sandi tidak berhasil	Kata sandi tidak berhasil diubah dan muncul	Konfirmasi kata sandi masih kosong.	Sesuai.

							diubah.	notifikasi konfirmasi kata sandi harus diisi.		
TC71	Skenario 32 : BF A2 A5	rizkariz20@gmail.com	email	261294	A123456	IN1234	Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata sandi belum sesuai.	Konfirmasi kata sandi belum sesuai dengan kata sandi baru.	Sesuai.
TC72	Skenario 32 : BF A2 A5	rizkariz20@gmail.com	email	261294	A123456		Kata sandi tidak berhasil diubah.	Kata sandi tidak berhasil diubah dan muncul notifikasi konfirmasi kata sandi harus diisi.	Konfirmasi kata sandi masih kosong.	Sesuai.

5.2.1.8 Penerapan *Manual Testing* Untuk *Use Case* Melihat Status Pengajuan Mobil

Berikut ini merupakan nilai aktual yang dirancang sesuai dengan matriks skenario yang telah dibuat di Bab 4. Nilai-nilai aktual dicantumkan pada Tabel 9 untuk *use case* melihat status pengajuan mobil. Nilai aktual diterapkan pada pengujian manual maupun otomatis. Pengujian dilakukan 3 kali dengan 3 variasi skenario yang berbeda-beda dengan tujuan untuk menemukan kesalahan yang mungkin terjadi. Terdapat skenario positif dan negatif dalam pembuatan nilai aktual. Pengujian manual *use case* ini dilakukan pada aplikasi acc.one dengan versi *update* tanggal 12 Februari 2020 dengan total waktu pengujian 1 menit 57 detik. Berdasarkan perbandingan *expected output* dan *output* yang dijabarkan pada Tabel 9, dapat ditarik kesimpulan bahwa pengujian *use case* melihat status pengajuan mobil secara manual memiliki persentase kesesuaian fungsionalitas 100%, yang berarti keluaran dari sistem semuanya sesuai dengan harapan awal.

Tabel 9. Nilai Aktual *Test Case* Untuk *Use Case* Melihat Status Pengajuan Mobil.

<i>Test Case ID</i>	Skenario	namaMobil	<i>Expected Output</i>	<i>Output</i>	Keterangan	Hasil Akhir
TC1	Skenario 1: BF	Agya	Status pengajuan dan detail informasi pengajuan mobil berhasil ditampilkan.	Status pengajuan dan detail informasi pengajuan mobil berhasil ditampilkan.	Menampilkan informasi detail pengajuan mobil.	Sesuai.
TC2	Skenario 2: BF A1	Calya	Status pengajuan dan detail informasi pengajuan mobil berhasil ditampilkan.	Status pengajuan dan detail informasi pengajuan mobil berhasil ditampilkan.	Menampilkan informasi detail pengajuan mobil.	Sesuai.
TC3	Skenario 3: BF A1 A2	Brio	Pengajuan mobil tidak berhasil ditampilkan.	Tidak ada notifikasi pengajuan tidak ditemukan dan status pengajuan dan detail informasi pengajuan mobil tidak berhasil	Pengajuan mobil tidak ditemukan.	Sesuai.

				ditampilkan.		
--	--	--	--	--------------	--	--



5.2.1.9 Penerapan *Manual Testing* Untuk *Use Case* Simulasi Kredit Mobil Baru (Paket)

Berikut ini merupakan nilai aktual yang dirancang sesuai dengan matriks skenario yang telah dibuat di Bab 4. Nilai-nilai aktual dicantumkan pada Tabel 10 untuk *use case* simulasi kredit mobil baru (paket). Nilai aktual diterapkan pada pengujian manual maupun otomatis. Pengujian dilakukan 4 kali dengan 4 variasi skenario yang berbeda-beda dengan tujuan untuk menemukan kesalahan yang mungkin terjadi. Terdapat skenario positif dan negatif dalam pembuatan nilai aktual. Pengujian manual *use case* ini dilakukan pada aplikasi acc.one dengan versi *update* tanggal 12 Februari 2020 dengan total waktu pengujian 3 menit 17 detik. Berdasarkan perbandingan *expected output* dan *output* yang dijabarkan pada Tabel 10, dapat ditarik kesimpulan bahwa pengujian *use case* simulasi kredit mobil baru (paket) secara manual memiliki persentase kesesuaian fungsionalitas 100%, yang berarti keluaran dari sistem semuanya sesuai dengan harapan awal.

Tabel 10. Nilai Aktual *Test Case* Untuk *Use Case* Simulasi Kredit Mobil Baru (Paket).

<i>Test Case ID</i>	<i>Skenario</i>	<i>namaMobil</i>	<i>persenDP</i>	<i>periodeBayar</i>	<i>creditOrCash</i>	<i>protection</i>	<i>Expected Output</i>	<i>Output</i>	<i>Keterangan</i>	<i>Hasil Akhir</i>
TC1	Skenario 1: BF	TOYOTA AGYA 2019 BENSIN	16	24	<i>Credit</i>	<i>No</i>	Sistem menampilkan jumlah angsuran dan total DP. Simulasi perhitungan kredit mobil baru paket berhasil disimpan.	Simulasi perhitungan kredit mobil tersimpan.	Menyimpan simulasi perhitungan kredit mobil baru paket.	Sesuai.

TC2	Skenario 2: BF A1	TOYOTA AGYA 2019 BENSIN	16	24	<i>Credit</i>	<i>Yes</i>	Sistem menampilkan jumlah angsuran dan total DP. Simulasi perhitungan kredit mobil baru paket berhasil disimpan.	Simulasi perhitungan kredit mobil tersimpan.	Sistem menambahkan jumlah angsuran per bulan.	Sesuai.
TC3	Skenario 3: BF A1 E1	TOYOTA AGYA 2019 BENSIN	25	48	<i>Cash</i>	<i>Yes</i>	Simulasi perhitungan kredit mobil baru paket batal disimpan.	Simulasi perhitungan kredit mobil batal tersimpan.	<i>Customer</i> batal menyimpan simulasi.	Sesuai.
TC4	Skenario 4: BF E1	TOYOTA AGYA 2019 BENSIN	25	48	<i>Cash</i>	<i>No</i>	Simulasi perhitungan kredit mobil baru paket batal disimpan.	Simulasi perhitungan kredit mobil batal tersimpan.	<i>Customer</i> memilih batal menyimpan simulasi.	Sesuai.

5.2.1.10 Penerapan *Manual Testing* Untuk *Use Case* Melihat Paket Pembelian Mobil

Berikut ini merupakan nilai aktual yang dirancang sesuai dengan matriks skenario yang telah dibuat di Bab 4. Nilai-nilai aktual dicantumkan pada Tabel 11 untuk *use case* melihat paket pembelian mobil. Nilai aktual diterapkan pada pengujian manual maupun otomatis. Pengujian dilakukan 1 kali dengan 1 variasi skenario yang berbeda-beda dengan tujuan untuk menemukan kesalahan yang mungkin terjadi. Terdapat skenario positif dan negatif dalam pembuatan nilai aktual. Pengujian manual *use case* ini dilakukan pada aplikasi acc.one dengan versi *update* tanggal 12 Februari 2020 dengan total waktu pengujian 24 detik. Berdasarkan perbandingan *expected output* dan *output* yang dijabarkan pada Tabel 11, dapat ditarik kesimpulan bahwa pengujian *use case* melihat paket pembelian mobil secara manual memiliki persentase kesesuaian fungsionalitas 100%, yang berarti keluaran dari sistem semuanya sesuai dengan harapan awal.

Tabel 11. Nilai Aktual *Test Case* Untuk *Use Case* Melihat Paket Pembelian Mobil.

<i>Test Case ID</i>	<i>Skenario</i>	<i>namaPaket</i>	<i>Expected Output</i>	<i>Output</i>	<i>Keterangan</i>	<i>Hasil Akhir</i>
TC1	Skenario 1: BF	ACC.ONE SEAMLESS ZENN	Informasi seputar paket berhasil ditampilkan.	Detail informasi paket pembelian mobil berhasil ditampilkan.	Menampilkan informasi seputar paket pembelian mobil baru.	Sesuai.

5.2.2 Analisis Hasil Pengujian Otomatis

5.2.2.1 Penerapan *Automated Testing* Pada Katalon Untuk *Use Case* Mencari Mobil Baru

Item	Object	Input	Output	Description
→ 1 - Start Application		"D:\MY WORLD\KULIAH (TUGAS)\TGS		
→ 2 - Delay		20		
→ 3 - Tap	boxCarriMobil	0		
→ 4 - Tap	boxMobilBaru	0		
5 - If Statement		brand == ""		
→ 5.1 - Tap	boxSearchMobilBaru	0		
→ 5.2 - Set Text	boxSearchMobilBaru	namaMobil; 0		
→ 5.3 - Tap	boxSearchMobilBaru	0		
→ 5.4 - Tap At Position		622; 1230		
→ 5.5 - Scroll To Text		specific		
→ 5.6 - Tap	tapObject	0		
5.7 - If Statement		condition == "mainPassed"		
→ 5.7.1 - Verify Element	textCautionMobilNotFoundBaru	0		
5.8 - Else Statement				
→ 5.8.1 - Verify Element	textCautionMobilNotFoundBaru	0		
6 - Else If Statement		brand == "daihatsu"		
→ 6.1 - Tap	boxDaihatsu	0		
6.2 - If Statement		merek != ""		
→ 6.2.1 - Tap	tapObject	0		
→ 6.2.2 - Scroll To Text		specific		
→ 6.2.3 - Tap	tapObject	0		

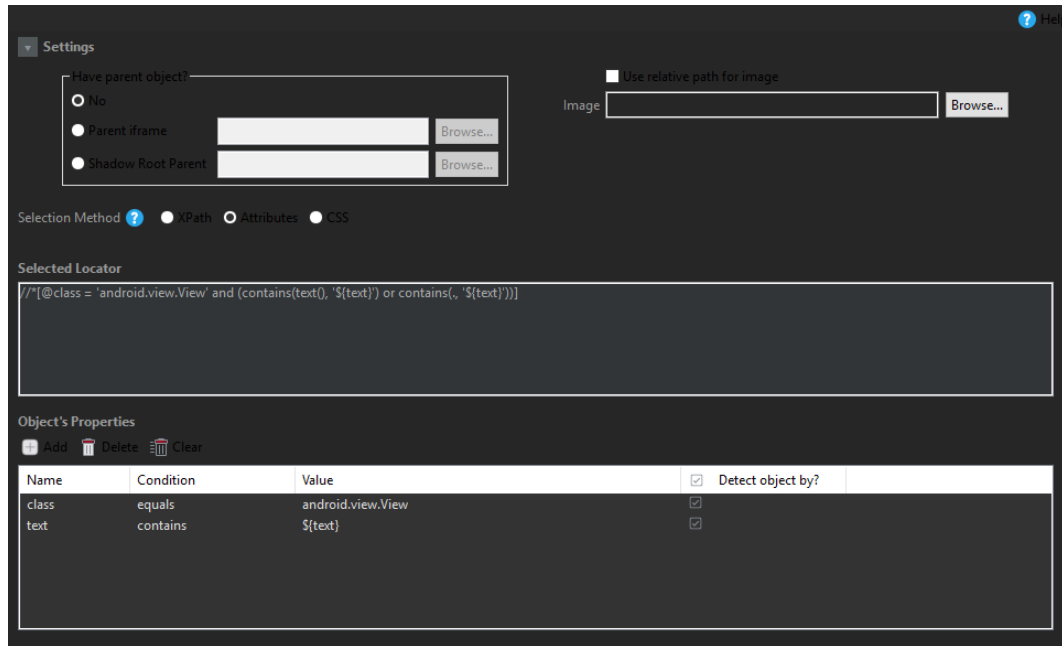
Gambar 18. *Test Case* Katalon Untuk *Use Case* Mencari Mobil Baru (1).

Item	Object	Input	Output	Description
→ 6.2.3 - Tap	tapObject	0		
6.3 - Else Statement				
→ 6.3.1 - Tap	searchBoxDaihatsu	0		
→ 6.3.2 - Set Text	searchBoxDaihatsu	namaMobil; 0		
→ 6.3.3 - Tap	searchBoxDaihatsu	0		
→ 6.3.4 - Tap At Position		622; 1230		
→ 6.3.5 - Scroll To Text		specific		
→ 6.3.6 - Tap	tapObject	0		
6.3.7 - If Statement		condition == "daihatsuPassed"		
→ 6.3.7.1 - Verify Element	textCautionMobilNotFoundBaru	0		
6.3.8 - Else Statement				
→ 6.3.8.1 - Verify Element	textCautionMobilNotFoundBaru	0		
7 - Else If Statement		brand == "toyota"		
→ 7.1 - Tap	boxToyota	0		
7.2 - If Statement		merek != ""		
→ 7.2.1 - Tap	tapObject	0		
→ 7.2.2 - Scroll To Text		specific		
→ 7.2.3 - Tap	tapObject	0		
7.3 - Else Statement				
→ 7.3.1 - Tap	searchBoxToyota	0		
→ 7.3.2 - Set Text	searchBoxToyota	namaMobil; 0		

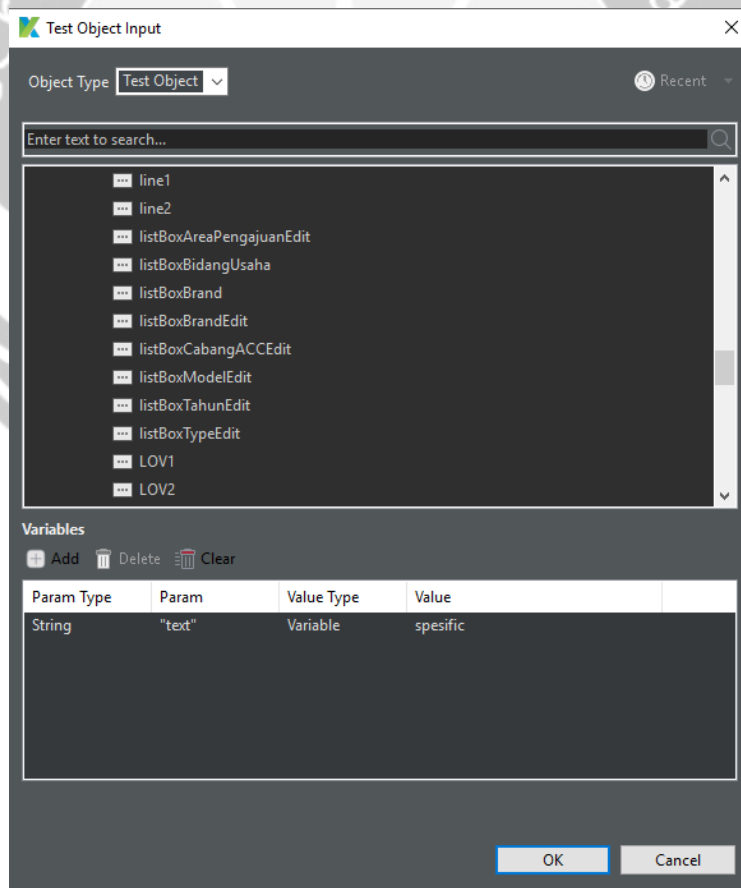
Gambar 19. *Test Case* Katalon Untuk *Use Case* Mencari Mobil Baru (2).

Item	Object	Input	Output	Description
→ 6.3.7.1 - Verify Element	textCautionMobilNotFoundBaru	0		
6.3.8 - Else Statement				
→ 6.3.8.1 - Verify Element	textCautionMobilNotFoundBaru	0		
7 - Else If Statement		brand == "toyota"		
→ 7.1 - Tap	boxToyota	0		
7.2 - If Statement		merek != ""		
→ 7.2.1 - Tap	tapObject	0		
→ 7.2.2 - Scroll To Text		specific		
→ 7.2.3 - Tap	tapObject	0		
7.3 - Else Statement				
→ 7.3.1 - Tap	searchBoxToyota	0		
→ 7.3.2 - Set Text	searchBoxToyota	namaMobil; 0		
→ 7.3.3 - Tap	searchBoxToyota	0		
→ 7.3.4 - Tap At Position		622; 1230		
→ 7.3.5 - Scroll To Text		specific		
→ 7.3.6 - Tap	tapObject	0		
7.3.7 - If Statement		condition == "toyotaPassed"		
→ 7.3.7.1 - Verify Element	textCautionMobilNotFoundBaru	0		
7.3.8 - Else Statement				
→ 7.3.8.1 - Verify Element	textCautionMobilNotFoundBaru	0		

Gambar 20. *Test Case* Katalon Untuk *Use Case* Mencari Mobil Baru (3).



Gambar 21. Kustomisasi Pembuatan Objek Baru.



Gambar 22. Salah Satu Contoh Penggunaan Objek Baru Pada Variabel Spesific.

Pada Gambar 18, 19, dan 20 merupakan rancangan *test case* untuk menguji fungsi mencari mobil baru. Fungsi ini diuji pada aplikasi acc.one dengan versi *update* per tanggal 21 Januari 2020. Pada dasarnya, fungsi mencari mobil baru ini tergolong ke dalam *use case* mengelola pembelian mobil *online* yang dijabarkan dalam *use case diagram* pada Gambar 3. Alur yang diterapkan pada pembuatan *test case* di Katalon sama dengan alur ketika pengujian manual. Pada pembuatan *test case* di Katalon, telah disediakan banyak instruksi seperti *tap* untuk melakukan aksi *tap* pada suatu objek, *set text* untuk memasukkan nilai string ke dalam suatu objek seperti *text field*, dan masih banyak instruksi lainnya.

Terdapat beberapa langkah dalam pembuatan *test case* yang mengharuskan untuk membuat objek baru, seperti objek `tapObject`, dikarenakan objek tersebut tidak dapat dideteksi jika dideteksi menggunakan Katalon dengan beberapa sebab seperti tidak adanya *resource-id* dari objek tersebut. Nantinya, objek tersebut akan dapat berperan sebagai objek yang mengandung teks dari nilai aktual variabel yang disesuaikan. Misalnya, `tapObject` di atas akan dapat dideteksi sebagai objek yang mengandung teks dengan nilai aktual dari variabel *specific*, yang berarti jika ada instruksi *tap* dengan objek `tapObject` dengan variabel *specific*, maka pada saat pengujian otomatis, sistem akan mencari objek dengan teks yang sesuai dengan nilai dari variabel *specific* yang telah diatur, kemudian akan *ditap*. Pengaturan pembuatan objek baru sama dengan Gambar 19 dan 20 dengan variabel yang disesuaikan dengan kebutuhan. *Tap At Position* pada *test case* ini sengaja digunakan juga sebagai alternatif apabila objek tidak dapat terdeteksi, sehingga sistem secara otomatis *tap* pada posisi yang telah ditentukan. Namun, kesulitannya terdapat pada ukuran layar perangkat pendukung pengujian seperti *smartphone* yang berbeda-beda sehingga, memungkinkan untuk mengganti-ganti posisi x dan y, nantinya jika jenis perangkatnya berbeda. *Verify Element Visible* pada *test case* ini berfungsi untuk mengecek apakah keluaran akan sesuai dengan yang diharapkan seperti *warning*, *alert*, atau notifikasi ketika kita melakukan suatu aksi.

No.	Name	Type	Default value
1	namaMobil	String	""
2	condition	String	""
3	spesific	String	""
4	merek	String	""
5	brand	String	""

Gambar 23. Variabel Katalon Untuk *Use Case* Mencari Mobil Baru.

Pada Gambar 21, ditunjukkan variabel-variabel yang digunakan pada pengujian otomatis di Katalon untuk *use case* mencari mobil baru. Variabel *condition* digunakan sebagai alat untuk membantu menentukan alur alternatif, sehingga alur pengujian dapat lebih singkat. Variabel *condition* juga dapat digunakan untuk mengurangi penggunaan *if-else* di dalam *if-else*. Dengan membuat kondisi yang lebih bervariasi, pembuatan *test case* nantinya akan lebih singkat dan mudah dipahami dengan mengandalkan nilai dari variabel *condition* sebagai kondisi di *if-else* dalam penentuan alur pengujian.

No.	ID	Description	Run
1	Test Cases/Cari Mobil/Baru Campur		

No.	ID	Data Iteration	Type
1	Data Files/Cari Mobil/cariMobilBaruCampur	All	One

No.	Name	Default value	Type	Test Data	Value
1	namaMobil	""	Data Column	1 - Data Files/Cari	namaMobil
2	condition	""	Data Column	1 - Data Files/Cari	condition
3	spesific	""	Data Column	1 - Data Files/Cari	spesific
4	merek	""	Data Column	1 - Data Files/Cari	merek
5	brand	""	Data Column	1 - Data Files/Cari	brand

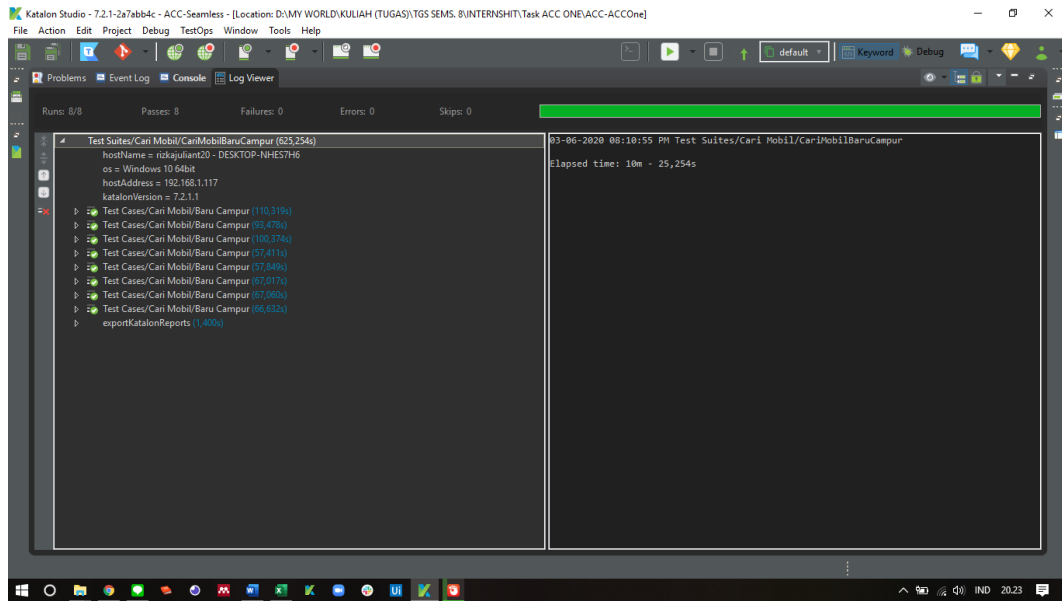
Gambar 24. *Test Suite* Katalon Untuk *Use Case* Mencari Mobil Baru.

Test suite yang ditunjukkan pada Gambar 22, berfungsi untuk menjalankan *test case* mencari mobil baru yang telah dibuat sebelumnya. Pada *test suite*, variabel-variabel yang sudah diinisialisasi pada *test case*, dipasangkan dengan *data binding* pada Microsoft Excel, agar nilai yang dimasukkan pada saat eksekusi sesuai.

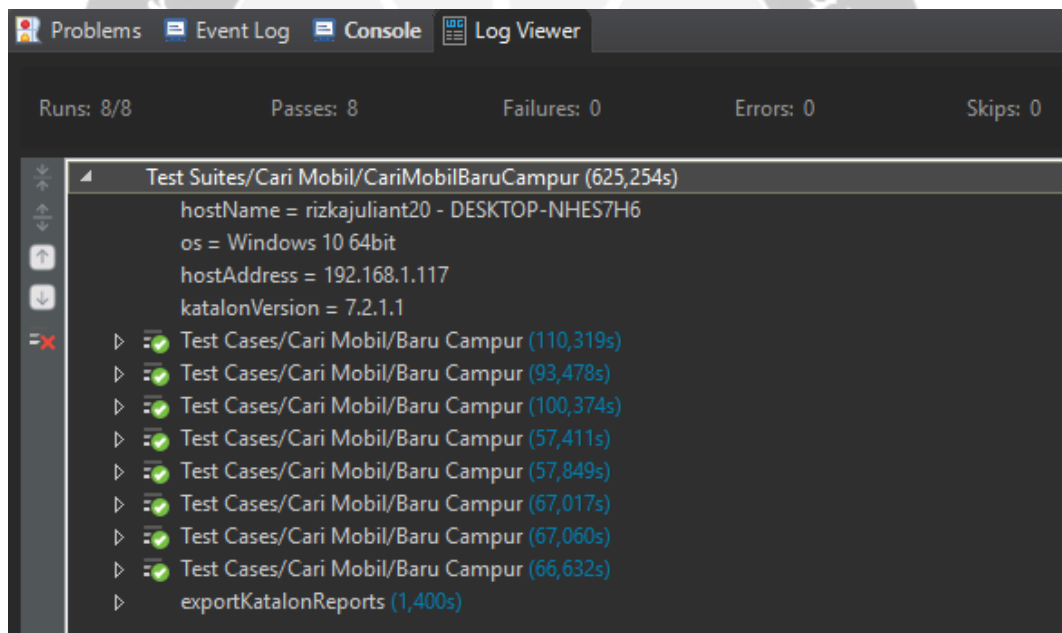
	A	B	C	D	E	F	G
1	namaMobil	condition	spesific	merek	brand		
2	Rush	mainPassed	Rush 1.5 G M/T				
3	Terios	daihatsuPassed	All New Terios X A/T		daihatsu		
4	Agya	toyotaPassed	Toyota Agya 1.0		toyota		
5		daihatsuPassed	All New Terios X A/T	All New Terios	daihatsu		
6		toyotaPassed	Toyota Agya 1.0	Agya	toyota		
7	Agya	daihatsuFailed			daihatsu		
8	Terios	toyotaFailed			toyota		
9	Brio	mainFailed					
10							

Gambar 25. *Data Binding* Untuk *Use Case* Mencari Mobil Baru.

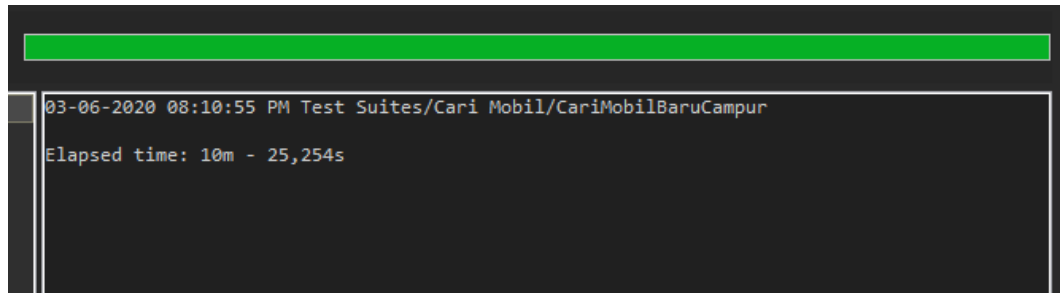
Pada Gambar 23, ditunjukkan *data binding* yang dibuat di Microsoft Excel, dengan nilai aktual yang disesuaikan dengan pengujian manual di mana *file* ini akan dimasukkan ke dalam Katalon. Terdapat 8 *data binding* yang berarti akan dilakukan 8 kali pengujian dengan nilai dan kondisi yang berbeda-beda untuk mendeteksi apakah kemungkinan positif dan negatifnya sudah sesuai dengan kriteria, sehingga diharapkan tidak terjadi *bug*. Pada *data binding* ini terdapat kondisi *mainPassed* di mana alur yang dijalankan akan mencapai *post condition*, *mainFailed* yakni dengan alur dasar namun nantinya tidak akan mencapai *post condition*. Adapun kondisi *daihatsuPassed* dan *toyotaPassed* tujuan yang dicapai akan sama dengan *mainPassed* hanya saja terdapat perbedaan di alur yang mana merujuk pada *alternate flow* untuk *use case* mencari mobil baru pada Bab 4. Sedangkan *daihatsuFailed* dan *toyotaFailed* adalah kondisi yang berbanding terbalik dengan *daihatsuPassed* dan *toyotaFailed*.



Gambar 26. Hasil Pengujian *Automated Testing* Untuk *Use Case* Mencari Mobil Baru (1).



Gambar 27. Hasil Pengujian *Automated Testing* Untuk *Use Case* Mencari Mobil Baru (2).

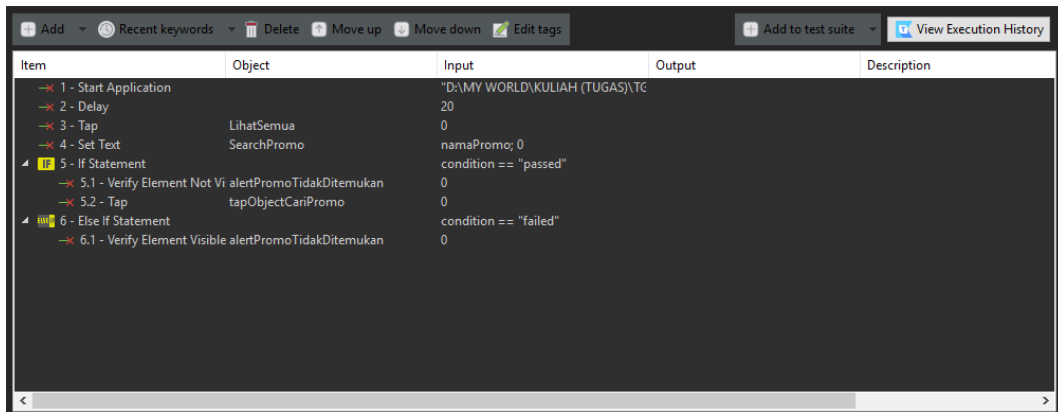
A screenshot of a terminal window with a black background and a green title bar. The text inside the terminal reads: "03-06-2020 08:10:55 PM Test Suites/Cari Mobil/CariMobilBaruCampur" followed by "Elapsed time: 10m - 25,254s".

```
03-06-2020 08:10:55 PM Test Suites/Cari Mobil/CariMobilBaruCampur
Elapsed time: 10m - 25,254s
```

Gambar 28. Hasil Pengujian *Automated Testing* Untuk *Use Case* Mencari Mobil Baru (3).

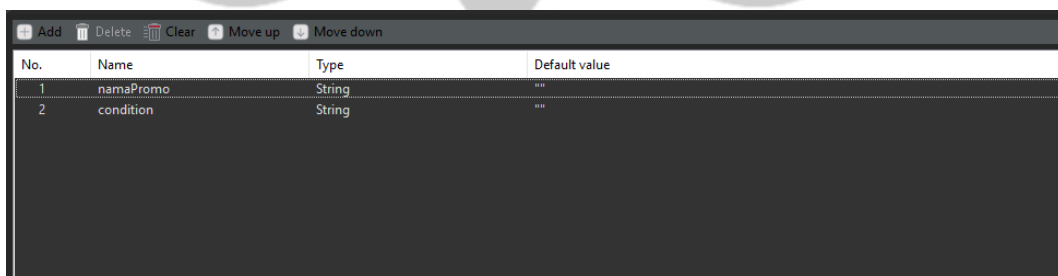
Pada Gambar 24, 25, dan 26, ditunjukkan hasil eksekusi pengujian otomatis *test case* mencari mobil baru. Waktu yang dibutuhkan untuk mengeksekusi *test suite* ini dengan 8 kali pengujian dan 8 *data binding* adalah 10 menit 25 detik. Hasil dari pengujian ini adalah *passed* dengan persentase kesesuaian fungsional 100% yang berarti dapat dikatakan tidak ditemukan *bug* pada fungsi yang diuji.

5.2.2.2 Penerapan *Automated Testing* Pada Katalon Untuk *Use Case* Mencari Promo Mobil Tertentu



Gambar 29. *Test Case* Katalon Untuk *Use Case* Mencari Promo Mobil Tertentu.

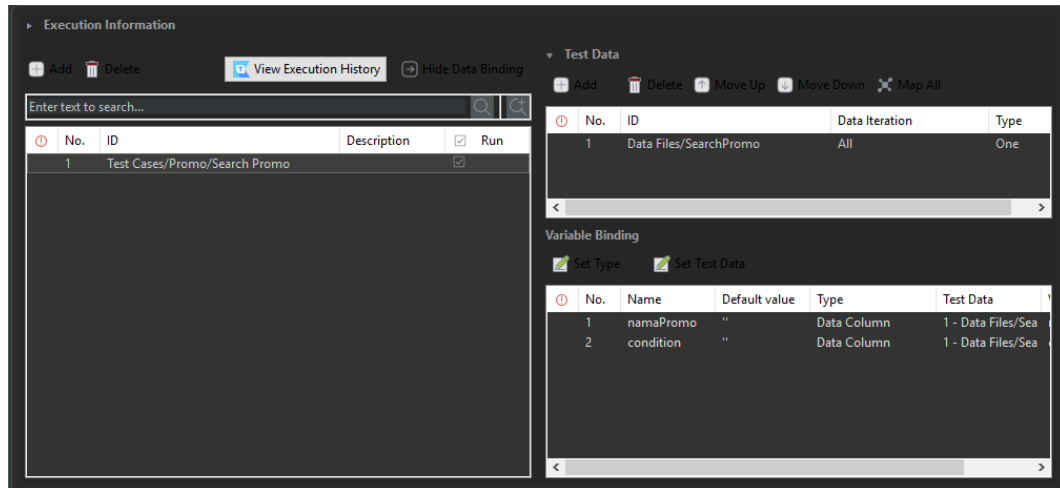
Pada Gambar 27, merupakan rancangan *test case* untuk menguji fungsi mencari promo mobil tertentu. Fungsi ini diuji pada aplikasi acc.one dengan versi *update* per tanggal 21 Januari 2020. Pada dasarnya, fungsi mencari promo mobil tertentu ini tergolong ke dalam *use case* mengelola pembelian mobil *online* yang dijabarkan dalam *use case diagram* pada Gambar 3. Alur pengujian yang diterapkan untuk *automated testing* ini sesuai dengan alur pengujian manual. Pada *test case*, dibuat objek baru untuk instruksi *tap* objek berdasarkan nilai variabel *namaPromo* yang telah dibuat di *data binding*. Pengaturan pembuatan objek baru sama dengan Gambar 19 dan 20 dengan variabel yang disesuaikan dengan kebutuhan.



Gambar 30. Variabel Katalon Untuk *Use Case* Mencari Promo Mobil Tertentu.

Pada Gambar 28, ditunjukkan variabel-variabel yang digunakan pada pengujian otomatis di Katalon untuk *use case* mencari promo mobil tertentu. Variabel *condition* digunakan sebagai alat untuk membantu menentukan alur alternatif, sehingga alur pengujian dapat lebih singkat. Variabel *condition* juga

dapat digunakan untuk mengurangi penggunaan *if-else* di dalam *if-else*. Dengan membuat kondisi yang lebih bervariasi, pembuatan *test case* nantinya akan lebih singkat dan mudah dipahami dengan mengandalkan nilai dari variabel condition sebagai kondisi di *if-else* dalam penentuan alur pengujian.



Gambar 31. Test Suite Katalon Untuk Use Case Mencari Promo Mobil Tertentu.

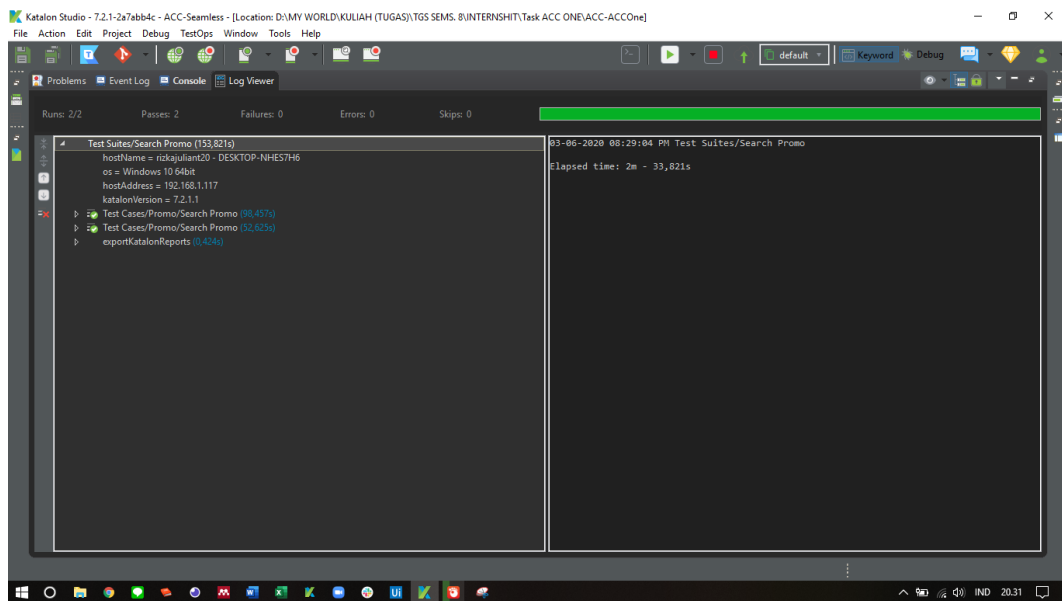
Test suite yang ditunjukkan pada Gambar 29, berfungsi untuk menjalankan *test case* mencari promo mobil tertentu yang telah dibuat sebelumnya. Pada *test suite*, variabel-variabel yang sudah diinisialisasi, dipasangkan dengan *data binding* pada Microsoft Excel, agar nilai yang dimasukkan pada saat pengujian sesuai.

	A	B	C	D	E	F	G	H	I	J	K
1	namaPromo	condition									
2	Paket Hemat	passed									
3	Hot Deals Mazda	failed									
4											
5											

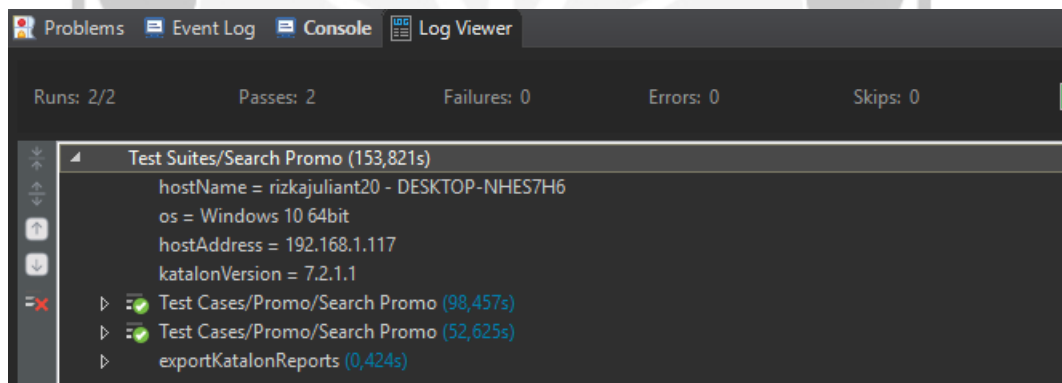
Gambar 32. Data Binding Untuk Use Case Mencari Promo Mobil Tertentu.

Pada Gambar 30, ditunjukkan *data binding* yang dibuat di Microsoft Excel, dengan nilai aktual yang disesuaikan dengan pengujian manual, di mana *file* ini akan dimasukkan ke dalam Katalon. Terdapat 2 *data binding* yang berarti akan dilakukan 2 kali pengujian dengan variasi skenario *testing* yang berbeda-beda untuk mendeteksi apakah terjadi *bug* dengan variasi skenario pengujian bernilai positif dan negatif. Pada *data binding* ini, terdapat kondisi passed, di

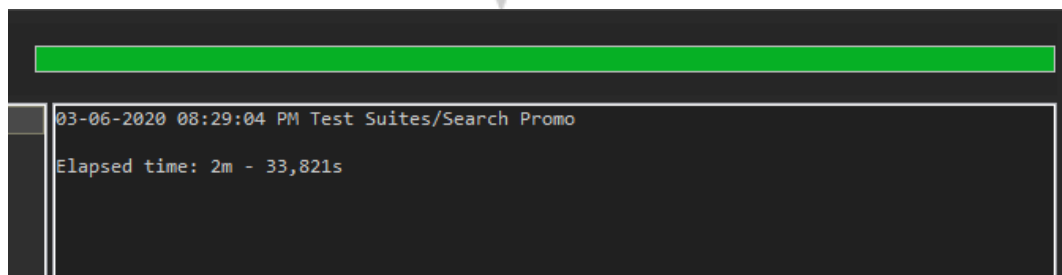
mana alur yang dijalankan akan mencapai *post condition*, sedangkan kondisi failed akan menjalankan sebaliknya.



Gambar 33. Hasil Pengujian *Automated Testing* Untuk *Use Case* Mencari Promo Mobil Tertentu (1).



Gambar 34. Hasil Pengujian *Automated Testing* Untuk *Use Case* Mencari Promo Mobil Tertentu (2).



Gambar 35. Hasil Pengujian *Automated Testing* Untuk *Use Case* Mencari Promo Mobil Tertentu (3).

Pada Gambar 31, 32, dan 33, ditunjukkan hasil eksekusi pengujian otomatis *test case* mencari promo mobil tertentu. Waktu yang dibutuhkan untuk mengeksekusi *test suite* ini dengan 2 kali pengujian dan 2 *data binding* adalah 2 menit 33 detik. Hasil dari keseluruhan pengujian ini adalah *passed* dengan persentase kesesuaian fungsional 100% di mana dapat dikatakan tidak ditemukan *bug* pada fungsi yang diuji.



5.2.2.3 Penerapan *Automated Testing* Pada Katalon Untuk *Use Case* Mencari Mobil Bekas (*Marketplace*)

Item	Object	Input	Output	Description
→ 1 - Start Application		"D:\MY WORLD\KULIAH (TUGAS)\TC		
→ 2 - Delay		20		
→ 3 - Tap	boxCariMobil	0		
→ 4 - Tap	boxMobilBekas	0		
→ 5 - Tap	boxMarketplace	0		
IF 6 - If Statement		brand == ""		
→ 6.1 - Tap	SearchBox	0		
→ 6.2 - Set Text	SearchBox	namaMobil; 0		
→ 6.3 - Tap	SearchBox	0		
→ 6.4 - Tap At Position		622; 1230		
→ 6.5 - Scroll To Text		specific		
→ 6.6 - Tap	tapObject	0		
IF 6.7 - If Statement		condition == "mainPassed"		
→ 6.7.1 - Verify Element N textCautionMobilNotFoundBaru		0		
ELSE 6.8 - Else Statement				
→ 6.8.1 - Verify Element V textCautionMobilNotFoundBaru		0		
ELSE 7 - Else If Statement		brand == "daihatsu"		
→ 7.1 - Tap	btnDaihatsu	0		
IF 7.2 - If Statement		merek != ""		
→ 7.2.1 - Swipe		641; 287; 200; 287		
→ 7.2.2 - Tap	tapObject	0		
→ 7.2.3 - Scroll To Text		specific		
→ 7.2.4 - Tap	tapObject	0		
ELSE 7.3 - Else Statement				
→ 7.3.1 - Tap	searchBoxToyota	0		
→ 7.3.2 - Set Text	searchBoxToyota	namaMobil; 0		

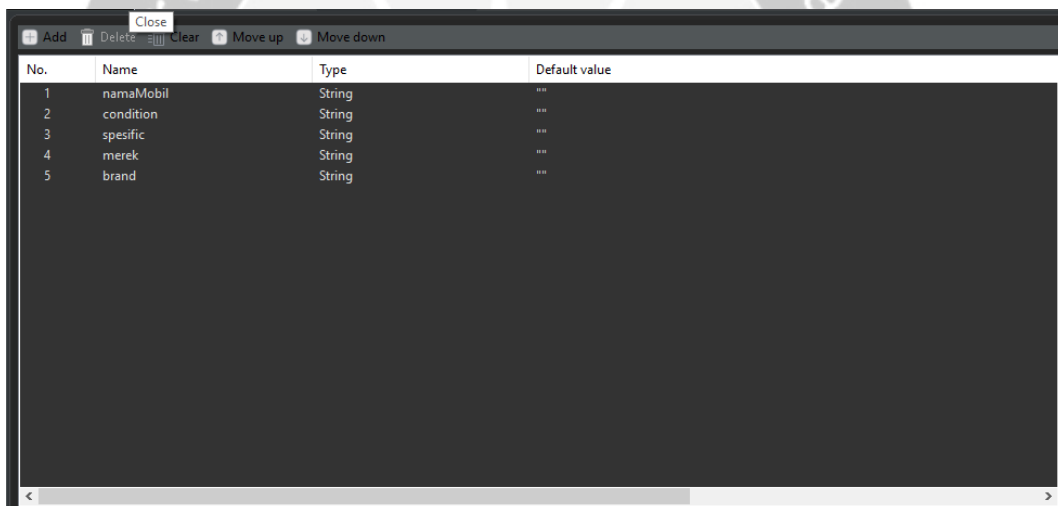
Gambar 36. *Test Case* Katalon Untuk *Use Case* Mencari Mobil Bekas (*Marketplace*) (1).

Item	Object	Input	Output	Description
→ 7.3.3 - Tap	searchBoxToyota	0		
→ 7.3.4 - Tap At Position		622; 1230		
→ 7.3.5 - Scroll To Text		specific		
→ 7.3.6 - Tap	tapObject	0		
IF 7.3.7 - If Statement		condition == "daihatsuPassed"		
→ 7.3.7.1 - Verify Element textCautionMobilNotFoundBaru		0		
ELSE 7.3.8 - Else Statement				
→ 7.3.8.1 - Verify Element textCautionMobilNotFoundBaru		0		
ELSE 8 - Else If Statement		brand == "toyota"		
→ 8.1 - Tap	btnToyota	0		
IF 8.2 - If Statement		merek != ""		
→ 8.2.1 - Tap	tapObject	0		
→ 8.2.2 - Scroll To Text		specific		
→ 8.2.3 - Tap	tapObject	0		
ELSE 8.3 - Else Statement				
→ 8.3.1 - Tap	searchBoxToyota	0		
→ 8.3.2 - Set Text	searchBoxToyota	namaMobil; 0		
→ 8.3.3 - Tap	searchBoxToyota	0		
→ 8.3.4 - Tap At Position		622; 1230		
→ 8.3.5 - Scroll To Text		specific		
→ 8.3.6 - Tap	tapObject	0		
IF 8.3.7 - If Statement		condition == "toyotaPassed"		
→ 8.3.7.1 - Verify Element textCautionMobilNotFoundBaru		0		
ELSE 8.3.8 - Else Statement				
→ 8.3.8.1 - Verify Element textCautionMobilNotFoundBaru		0		
→ 9 - Send Keys	SearchBox	Keys.chord(Keys.META)		

Gambar 37. *Test Case* Katalon Untuk *Use Case* Mencari Mobil Bekas (*Marketplace*) (2).

Pada Gambar 34 dan 35, merupakan rancangan *test case* untuk menguji fungsi mencari mobil bekas (*marketplace*). Fungsi ini diuji pada aplikasi acc.one dengan versi *update* per tanggal 21 Januari 2020. Pada dasarnya, fungsi mencari mobil bekas (*marketplace*) ini tergolong ke dalam *use case* mengelola pembelian

mobil *online* yang dijabarkan di *use case diagram* pada Gambar 3. Alur pengujian yang diterapkan untuk *automated testing* ini sesuai dengan alur pengujian manual. Pada *test case* terdapat *Tap At Position* yang berfungsi untuk *tap enter* pada poin 6.4, dikarenakan fungsi `sendKeys(Enter)` yang merupakan instruksi bawaan dari Katalon itu sendiri tidak dapat berfungsi. Terdapat objek baru yang dibuat yakni `tapObject` dengan tujuan untuk mendeteksi objek berdasarkan teks. Dalam *test case* ini, `tapObject` berperan sebagai objek yang mengandung teks sesuai dengan nilai aktual dari variabel merek dan spesifik. *VerifyElementVisible* merupakan instruksi untuk mendeteksi apakah terdapat objek yang diharapkan atau tidak, seperti salah satu langkah pada *test case* di atas, *VerifyElementVisible* berfungsi untuk mendeteksi adanya notifikasi mobil tidak ditemukan. Pengaturan pembuatan objek baru sama dengan Gambar 19 dan 20 dengan variabel yang disesuaikan dengan kebutuhan.

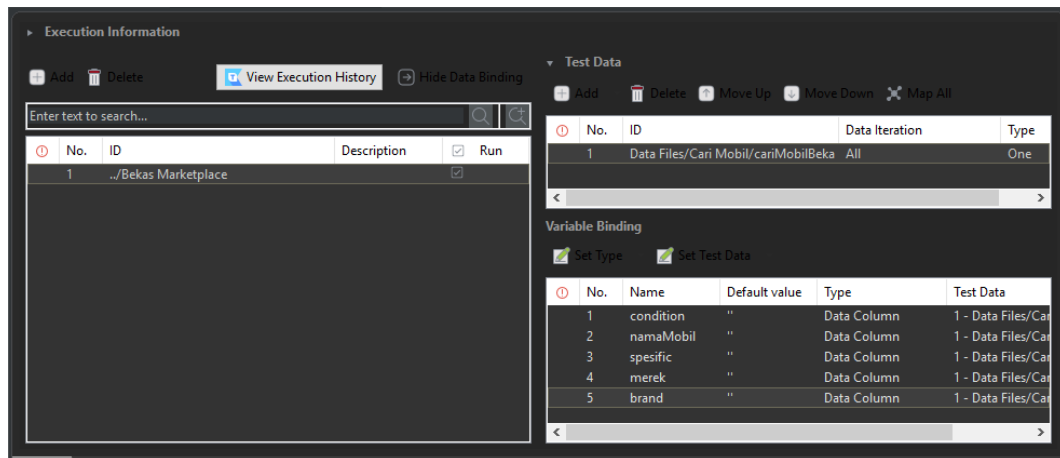


No.	Name	Type	Default value
1	namaMobil	String	''
2	condition	String	''
3	spesific	String	''
4	merek	String	''
5	brand	String	''

Gambar 38. Variabel Katalon Untuk *Use Case* Mencari Mobil Bekas (*Marketplace*).

Pada Gambar 36, ditunjukkan variabel-variabel yang digunakan pada pengujian otomatis di Katalon untuk *use case* mencari mobil bekas (*marketplace*). Variabel `condition` digunakan sebagai alat untuk membantu menentukan alur alternatif, sehingga alur pengujian dapat lebih singkat. Variabel `condition` juga dapat digunakan untuk mengurangi penggunaan *if-else* di dalam *if-else*. Dengan membuat kondisi yang lebih bervariasi, pembuatan *test case* nantinya akan lebih

singkat dan mudah dipahami dengan mengandalkan nilai dari variabel condition sebagai kondisi di *if-else* dalam penentuan alur pengujian.



Gambar 39. *Test Suite* Katalon Untuk *Use Case* Mencari Mobil Bekas (*Marketplace*).

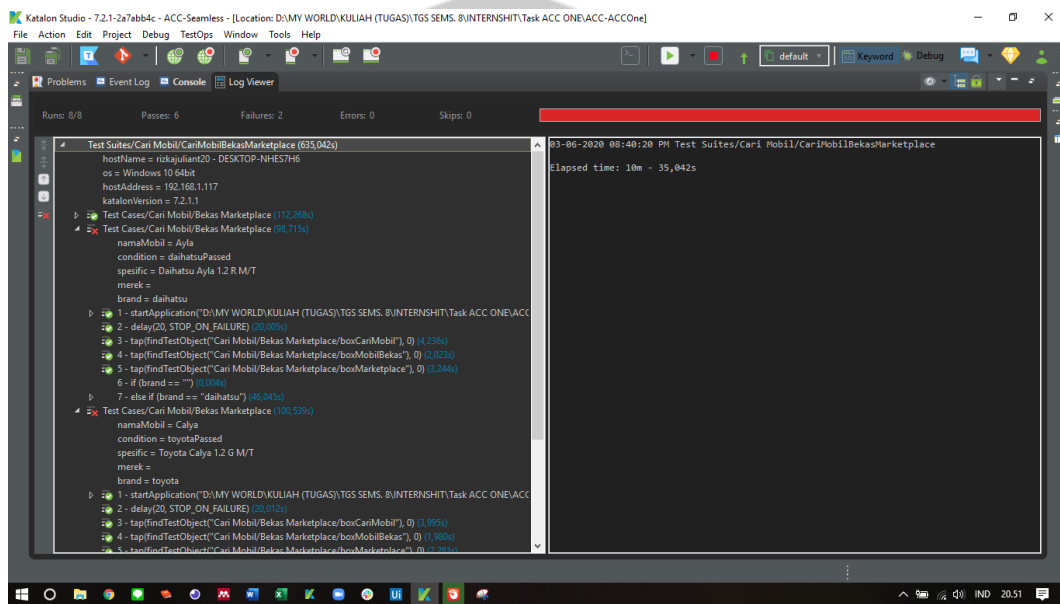
Test suite yang ditunjukkan pada Gambar 37, berfungsi untuk menjalankan *test case* mencari mobil bekas (*marketplace*) yang telah dibuat sebelumnya. Pada *test suite*, variabel-variabel yang sudah diinisialisasi, dipasangkan dengan *data binding* pada Microsoft Excel, agar nilai yang dimasukkan pada saat pengujian sesuai.

	A	B	C	D	E	F	G	H
1	namaMobil	condition	specific	merek	brand			
2	Xenia	mainPassed	Daihatsu Xenia 1.3 R CSTM A/T Bensin					
3	Ayla	daihatsuPassed	Daihatsu Ayla 1.2 R M/T		daihatsu			
4	Calya	toyotaPassed	Toyota Calya 1.2 G M/T		toyota			
5		daihatsuPassed	Daihatsu Ayla 1.2 R M/T	New Ayla	daihatsu			
6		toyotaPassed	Toyota Calya 1.2 G M/T	Calya	toyota			
7	Calya	daihatsuFailed			daihatsu			
8	Terios	toyotaFailed			toyota			
9	Brio	mainFailed						

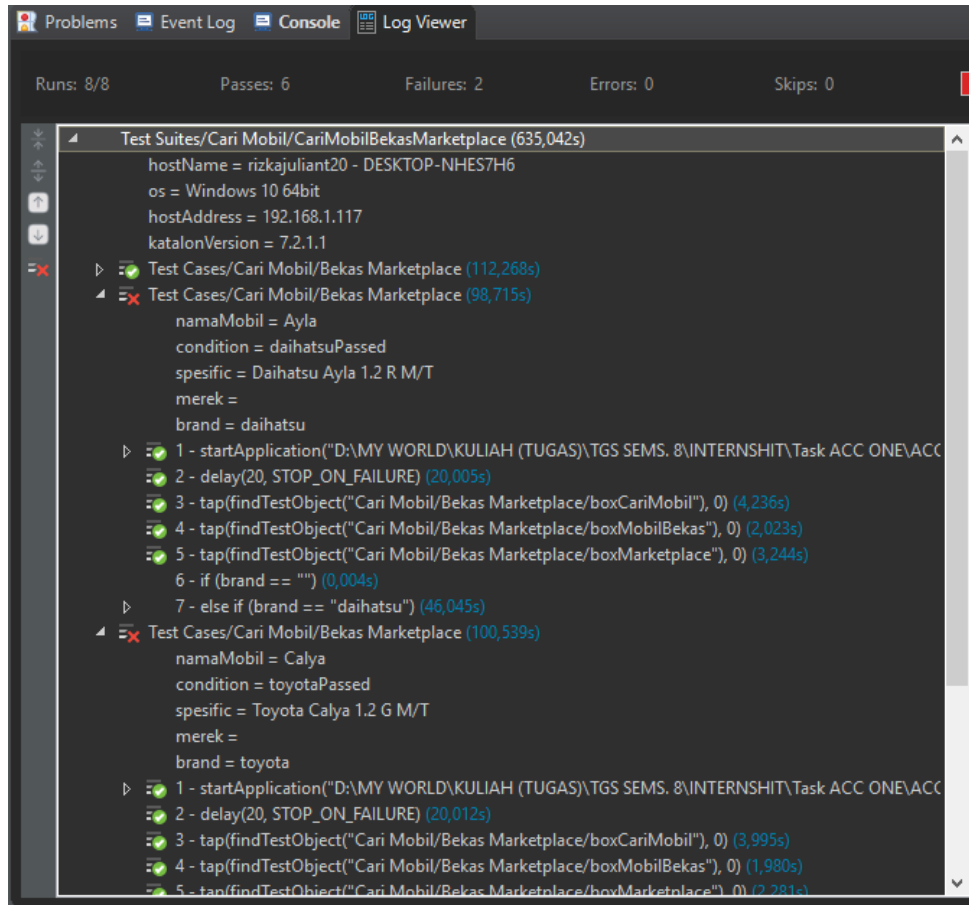
Gambar 40. *Data Binding* Untuk *Use Case* Mencari Mobil Bekas (*Marketplace*).

Pada Gambar 38, ditunjukkan *data binding* yang dibuat pada Microsoft Excel, dengan nilai aktual yang disesuaikan dengan pengujian manual, di mana *file* ini akan dimasukkan ke dalam Katalon. Terdapat 9 *data binding* yang berarti akan dilakukan 9 kali pengujian dengan variasi skenario *testing* yang berbeda-beda untuk mendeteksi apakah terjadi *bug* dengan variasi skenario pengujian bernilai positif dan negatif. Pada *data binding* ini terdapat kondisi *mainPassed* di

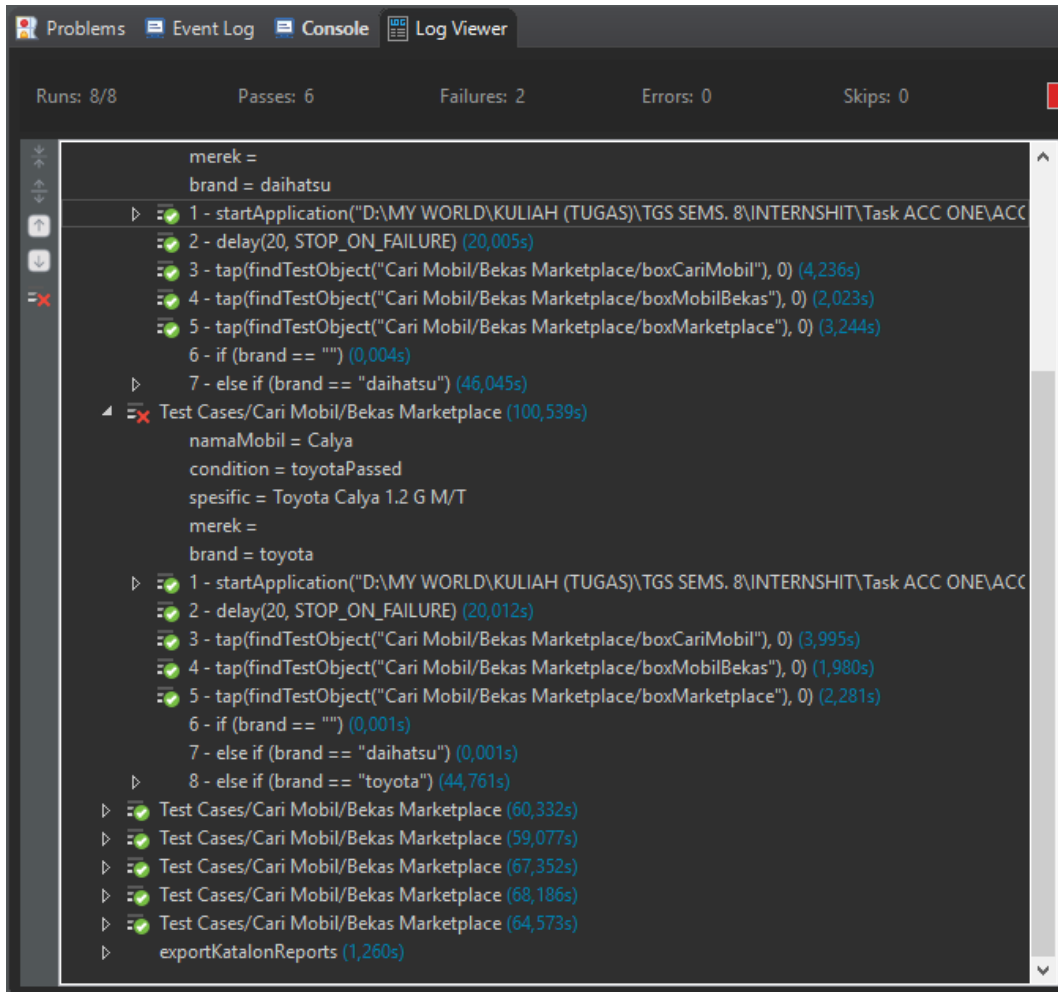
mana alur yang dijalankan akan mencapai *post condition*, mainFailed yakni dengan alur dasar namun nantinya tidak akan mencapai *post condition*. Adapun kondisi daihatsuPassed dan toyotaPassed tujuan yang dicapai akan sama dengan mainPassed hanya saja terdapat perbedaan di alur yang mana merujuk pada *alternate flow* untuk *use case* mencari mobil baru pada Bab 4. Sedangkan daihatsuFailed dan toyotaFailed adalah kondisi yang berbanding terbalik dengan daihatsuPassed dan toyotaFailed.



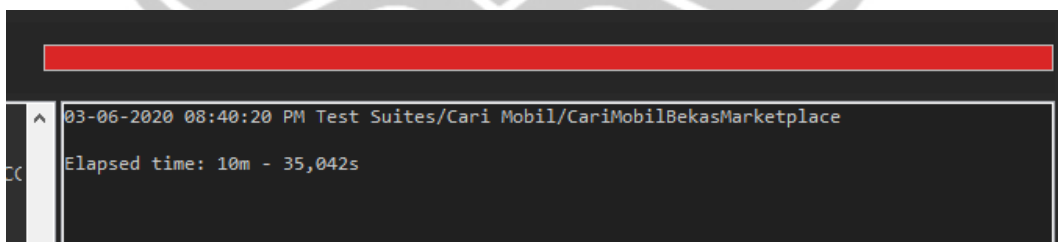
Gambar 41. Hasil Automated Testing Untuk Use Case Mencari Mobil Bekas (Marketplace) (1).



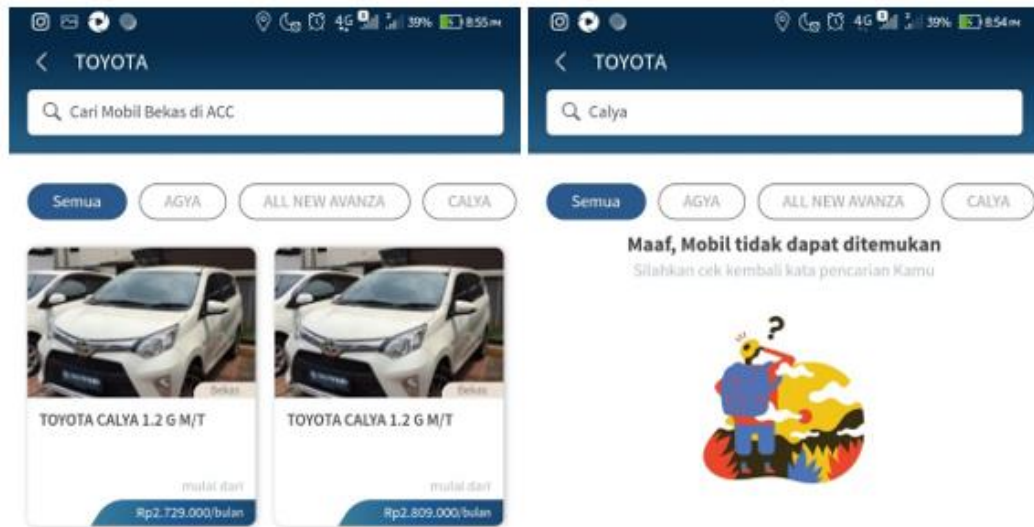
Gambar 42. Hasil Automated Testing Untuk Use Case Mencari Mobil Bekas (Marketplace) (2).



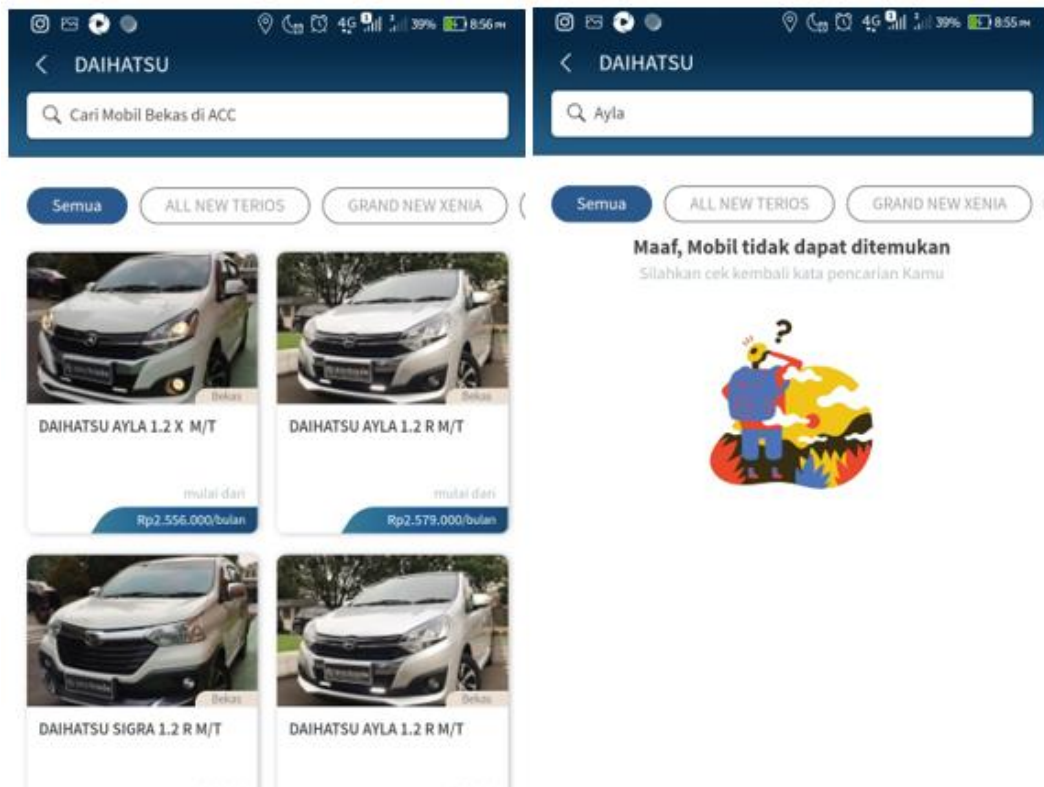
Gambar 43. Hasil *Automated Testing* Untuk *Use Case* Mencari Mobil Bekas (*Marketplace*) (3).



Gambar 44. Hasil *Automated Testing* Untuk *Use Case* Mencari Mobil Bekas (*Marketplace*) (4).



Gambar 45. Bug yang Ditemukan Pada Use Case Mencari Mobil Bekas (Marketplace) Brand Toyota.



Gambar 46. Bug yang Ditemukan Pada Use Case Mencari Mobil Bekas (Marketplace) Brand Daihatsu.

Pada Gambar 39, 40, 41, dan 42, ditunjukkan hasil eksekusi pengujian otomatis *test case* mencari promo mobil tertentu. Waktu yang dibutuhkan untuk mengeksekusi *test suite* ini dengan 9 kali pengujian dan 9 *data binding* adalah 10 menit 35 detik. Hasil dari keseluruhan pengujian ini adalah *failed* dengan persentase kesesuaian fungsional 75%, di mana 25% dari *data binding* gagal untuk dieksekusi yang menandakan adanya *bug*. Bukti adanya *bug* dalam aplikasi ditunjukkan pada Gambar 43 dan 44. Proses pengujian gagal ketika mengeksekusi data ke-2 dan ke-3 yang seharusnya mobil dapat ditemukan apabila dicari dengan cara *scroll* di halaman utama setelah masuk ke bagian *brand* tertentu, namun pada saat dicari dengan cara mengetikkan nama mobil di *text field* pencarian, hasil tidak ditemukan.



5.2.2.4 Penerapan *Automated Testing* Pada Katalon Untuk *Use Case* Mencari Mobil Bekas (Lelang)

Item	Object	Input	Output	Description
1 - Start Application		"D:\MY WORLD\KULIAH (TUGAS)\TC		
2 - Delay		20		
3 - Tap	boxCariMobil	0		
4 - Tap	boxMobilBekas	0		
5 - Tap	boxLelang	0		
6 - Delay		10		
7 - Swipe		660; 149; 70; 149		
8 - Delay		5		
9 - Tap	btnIBID	0		
10 - Delay		5		
11 - Tap	btnPilihLokasi	0		
12 - Hide Keyboard				
13 - If Statement		lokasi == "BATAM"		
13.1 - Tap	btnTapObject	0		
13.2 - Delay		5		
13.3 - Tap	btnTapObject	0		
13.4 - Delay		5		
13.5 - Tap	btnTapObject	0		
13.6 - Delay		5		
13.7 - Set Text	txtBoxCariMobil	namaMobil; 0		
13.8 - If Statement		condition == "passed"		
13.8.1 - Verify Element	alertCarNotFound	0		
13.8.2 - Tap	btnTapObject	0		

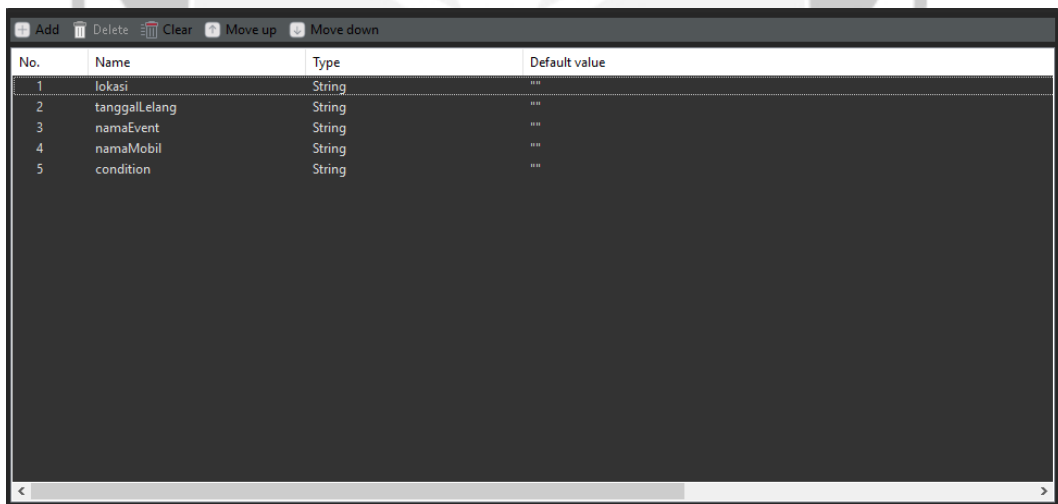
Gambar 47. *Test Case* Katalon Untuk *Use Case* Mencari Mobil Bekas (Lelang) (1).

Item	Object	Input	Output	Description
11 - Tap	btnPilihLokasi	0		
12 - Hide Keyboard				
13 - If Statement		lokasi == "BATAM"		
13.1 - Tap	btnTapObject	0		
13.2 - Delay		5		
13.3 - Tap	btnTapObject	0		
13.4 - Delay		5		
13.5 - Tap	btnTapObject	0		
13.6 - Delay		5		
13.7 - Set Text	txtBoxCariMobil	namaMobil; 0		
13.8 - If Statement		condition == "passed"		
13.8.1 - Verify Element	alertCarNotFound	0		
13.8.2 - Tap	btnTapObject	0		
13.9 - Else Statement				
14 - Else Statement				
14.1 - If Statement		lokasi == ""		
14.1.1 - Tap At Position		87; 419		
14.1.2 - Verify Element	alertHarusDipilih	0		
14.2 - Else Statement				
14.2.1 - Tap	btnTapObject	0		
14.3 - Tap	btnTapObject	0		
14.4 - Verify Element Visibl	alertTidakAdaEventLelang	0		

Gambar 48. *Test Case* Katalon Untuk *Use Case* Mencari Mobil Bekas (Lelang) (2).

Pada Gambar 45 dan 46, merupakan rancangan *test case* untuk menguji fungsi mencari mobil bekas (lelang). Fungsi ini diuji pada aplikasi acc.one dengan versi *update* per tanggal 21 Januari 2020. Pada dasarnya, fungsi ini tergolong ke dalam *use case* melakukan lelang mobil *online* yang dijabarkan dalam *use case diagram* pada Gambar 3. Alur pengujian yang diterapkan untuk *automated testing* ini sesuai dengan alur pengujian manual. Pada *test case* terdapat *Tap At Position*

yang berfungsi untuk *tap* layar selain bagian lokasi untuk menunjukkan bahwa muncul notifikasi bahwa lokasi harus dipilih. Terdapat objek baru yang dibuat yakni `btnTapObject` dengan tujuan untuk mendeteksi objek berdasarkan teks. Dalam *test case* ini, `btnTapObject` berperan sebagai objek yang mengandung teks sesuai dengan nilai aktual dari variabel lokasi, tanggalLelang, namaEvent dan namaMobil. *VerifyElementVisible* merupakan instruksi untuk mendeteksi apakah terdapat objek yang diharapkan atau tidak, seperti salah satu langkah pada *test case* di atas, *VerifyElementVisible* berfungsi untuk mendeteksi adanya notifikasi mobil tidak ditemukan ataupun tidak ada *event* lelang. Pengaturan pembuatan objek baru sama dengan Gambar 19 dan 20 dengan variabel yang disesuaikan dengan kebutuhan. Instruksi *Swipe* dalam *test case* kali ini digunakan untuk menggeser pilihan-pilihan *button* perusahaan yang harus dipilih, yang mana lokasi dari *button* perusahaan lelang IBID ada di paling akhir yang mengharuskan untuk menggeser dari posisi kanan ke kiri. Instruksi *Swipe* membutuhkan posisi x dan y awal, serta posisi x dan y akhir sebagai batasan seberapa jauh penggeseran dilakukan.

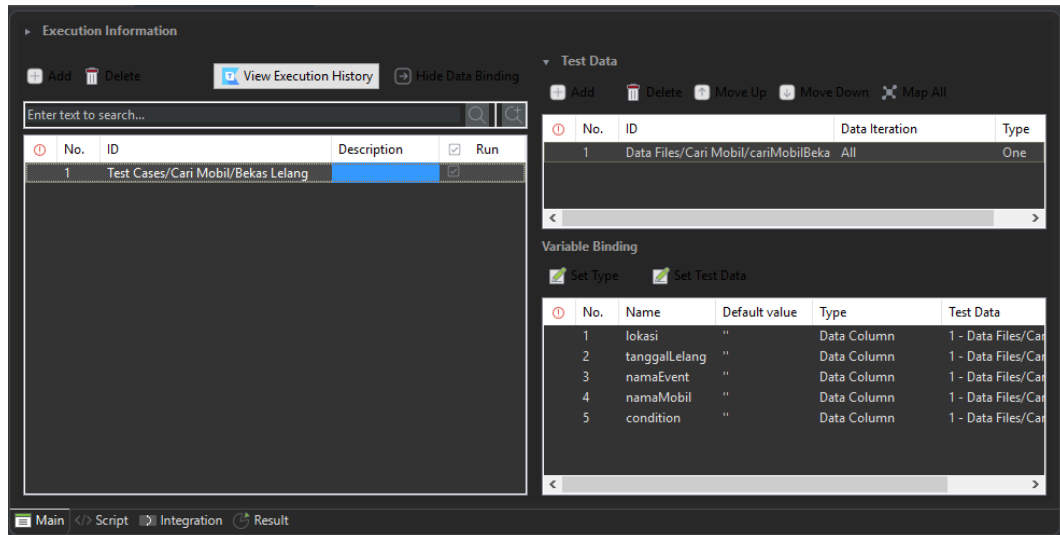


No.	Name	Type	Default value
1	lokasi	String	""
2	tanggalLelang	String	""
3	namaEvent	String	""
4	namaMobil	String	""
5	condition	String	""

Gambar 49. Variabel Katalon Untuk *Use Case* Mencari Mobil Bekas (Lelang).

Pada Gambar 47, ditunjukkan variabel-variabel yang digunakan pada pengujian otomatis di Katalon untuk *use case* mencari mobil bekas (lelang). Variabel *condition* digunakan sebagai alat untuk membantu menentukan alur alternatif, sehingga alur pengujian dapat lebih singkat. Variabel *condition* juga dapat digunakan untuk mengurangi penggunaan *if-else* di dalam *if-else*. Dengan

membuat kondisi yang lebih bervariasi, pembuatan *test case* nantinya akan lebih singkat dan mudah dipahami dengan mengandalkan nilai dari variabel condition sebagai kondisi di *if-else* dalam penentuan alur pengujian.



Gambar 50. *Test Suite* Katalon Untuk *Use Case* Mencari Mobil Bekas (Lelang).

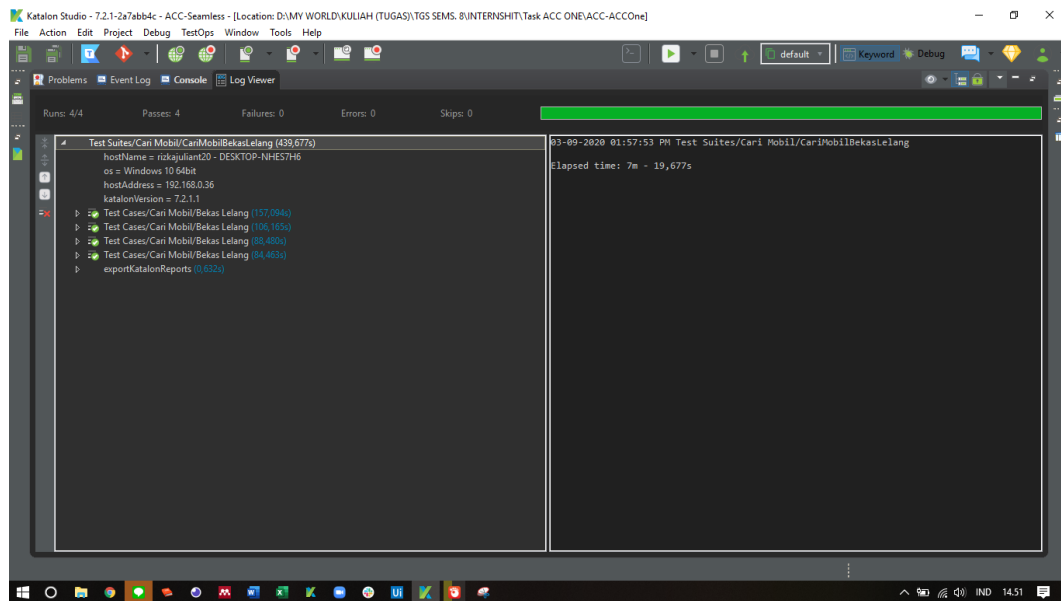
Test suite yang ditunjukkan pada Gambar 48, berfungsi untuk menjalankan *test case* mencari mobil bekas (lelang) yang telah dibuat sebelumnya. Pada *test suite*, variabel-variabel yang sudah diinisialisasi, dipasangkan dengan *data binding* pada Microsoft Excel, agar nilai yang dimasukkan pada saat pengujian sesuai.

	A	B	C	D	E	F	G	H	I
1	lokasi	tanggalLelang	namaEvent	namaMobil	condition				
2	BATAM	14	jangan hilang	Chery QQ GX	passed				
3	BATAM	17	jangan hilang	Agya	failed				
4	BALIKPAPAN	15			failed				
5		15			failed				

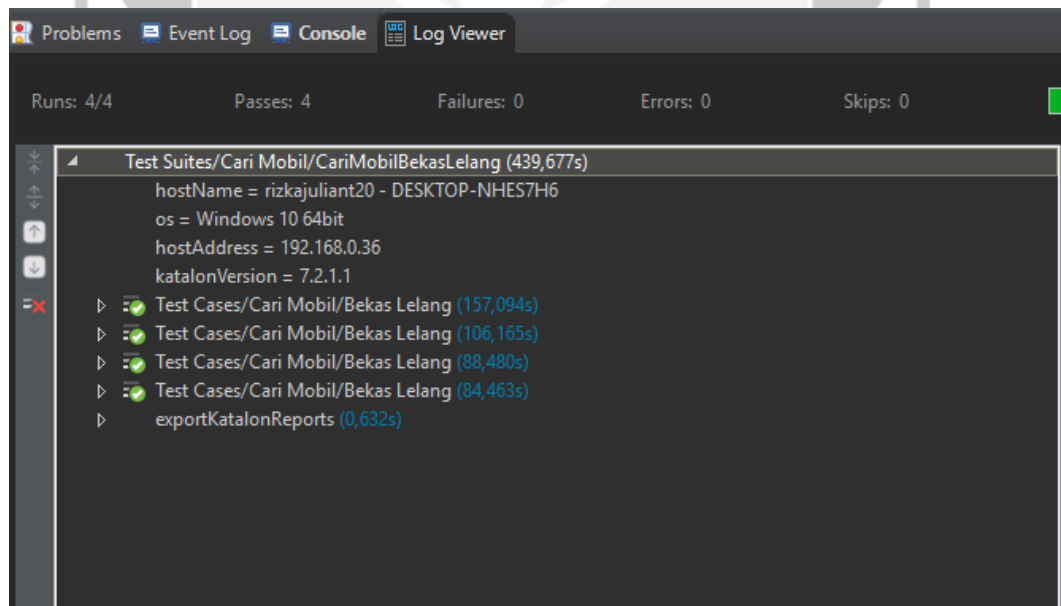
Gambar 51. *Data Binding* Untuk *Use Case* Mencari Mobil Bekas (Lelang).

Pada Gambar 49, ditunjukkan *data binding* yang dibuat pada Microsoft Excel, dengan nilai aktual yang disesuaikan dengan pengujian manual, di mana *file* ini akan dimasukkan ke dalam Katalon. Terdapat 5 *data binding* yang berarti akan dilakukan 5 kali pengujian dengan variasi skenario *testing* yang berbeda-beda untuk mendeteksi apakah terjadi *bug* dengan variasi skenario pengujian bernilai positif dan negatif. Pada *data binding* ini, terdapat kondisi passed, di

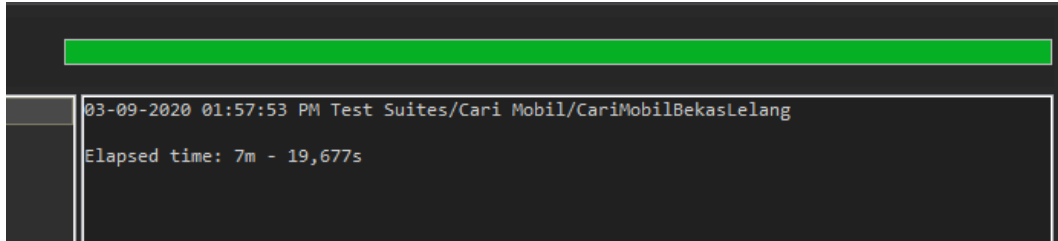
mana alur yang dijalankan akan mencapai *post condition*, sedangkan kondisi failed akan menjalankan sebaliknya.



Gambar 52. Hasil *Automated Testing* Untuk *Use Case* Mencari Mobil Bekas (Lelang) (1).



Gambar 53. Hasil *Automated Testing* Untuk *Use Case* Mencari Mobil Bekas (Lelang) (2).

A screenshot of a terminal window showing the results of an automated test. At the top, there is a green progress bar. Below it, the text reads: "03-09-2020 01:57:53 PM Test Suites/Cari Mobil/CariMobilBekasLelang" and "Elapsed time: 7m - 19,677s".

```
03-09-2020 01:57:53 PM Test Suites/Cari Mobil/CariMobilBekasLelang
Elapsed time: 7m - 19,677s
```

Gambar 54. Hasil *Automated Testing* Untuk *Use Case* Mencari Mobil Bekas (Lelang) (3).

Pada Gambar 50, 51, dan 52, ditunjukkan hasil eksekusi pengujian otomatis *test case* mencari mobil bekas (lelang). Waktu yang dibutuhkan untuk mengeksekusi *test suite* ini dengan 5 kali pengujian dan 5 *data binding* adalah 7 menit 19 detik. Hasil dari keseluruhan pengujian ini adalah *passed* dengan persentase kesesuaian fungsional 100% di mana dapat dikatakan tidak ditemukan *bug* pada fungsi yang diuji.

5.2.2.5 Penerapan *Automated Testing* Pada Katalon Untuk *Use Case* Mengubah Foto Profil

Item	Object	Input	
1 - Start Application		"D:\MY WORLD\KULIAH (TUGAS)\TGS SEMS. 8\INTERNSHIP\Backup\ACC-ACCOne\acc.one.apk"; false	0
2 - Delay		20	
3 - Tap	txtMyAccount	0	
4 - Delay		5	
5 - If Statement		picture == "galeri"	
5.1 - Tap	btnEditPhotoProfile	0	
5.2 - Tap	btnGallery	0	
5.3 - Delay		5	
5.4 - Scroll To Text		namaFoto	
5.5 - Tap	Select Photo	0	
5.6 - If Statement		condition == "passed"	
5.6.1 - Verify Element Visible	Foto Profile Berhasil Diganti	0	
6 - Else Statement			
6.1 - Tap	btnEditPhotoProfile	0	
6.2 - Tap	btnOptionCamera	0	
6.3 - Delay		5	
6.4 - Tap At Position		380; 1166	
6.5 - If Statement		condition == "passedRetake"	
6.6 - Delay		5	
6.7 - Tap	btnDone	0	
6.8 - Verify Element Visible	Foto Profile Berhasil Diganti	0	

Gambar 55. *Test Case* Katalon Untuk *Use Case* Mengubah Foto Profil.

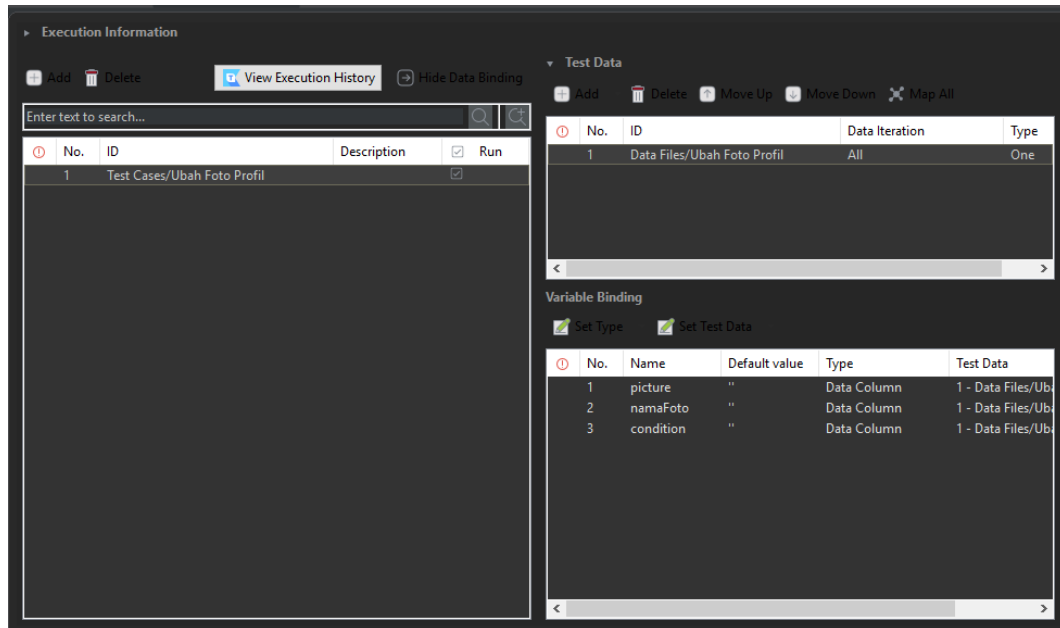
Pada Gambar 53, merupakan rancangan *test case* untuk menguji fungsi mengubah foto profil. Fungsi ini diuji pada aplikasi acc.one dengan versi *update* per tanggal 21 Januari 2020. Pada dasarnya, fungsi mengubah foto profil ini tergolong ke dalam *use case* mengelola data akun *customer* yang dijabarkan pada Gambar 3. Alur pengujian yang diterapkan untuk *automated testing* ini sesuai dengan alur pengujian manual. Pada *test case*, terdapat 2 alur utama di mana dapat mengganti foto profil dengan cara memilih dari galeri atau membuka kamera perangkat dan mengambil foto secara langsung. Terdapat pula instruksi *Tap At Position* pada *test case* yang berfungsi untuk *tap button* ambil gambar seperti yang ditunjukkan pada poin 6.4 dan ambil gambar ulang pada poin 6.5.3.

No.	Name	Type	Default value
1	picture	String	""
2	namaFoto	String	""
3	condition	String	""

Gambar 56. Variabel Katalon Untuk *Use Case* Mengubah Foto Profil.

Pada Gambar 54, ditunjukkan variabel-variabel yang digunakan pada pengujian otomatis di Katalon untuk *use case* mengubah foto profil. Variabel

condition digunakan sebagai alat untuk membantu menentukan alur alternatif, sehingga alur pengujian dapat lebih singkat. Variabel condition juga dapat digunakan untuk mengurangi penggunaan *if-else* di dalam *if-else*. Dengan membuat kondisi yang lebih bervariasi, pembuatan *test case* nantinya akan lebih singkat dan mudah dipahami dengan mengandalkan nilai dari variabel condition sebagai kondisi di *if-else* dalam penentuan alur pengujian.



Gambar 57. *Test Suite* Katalon Untuk *Use Case* Mengubah Foto Profil.

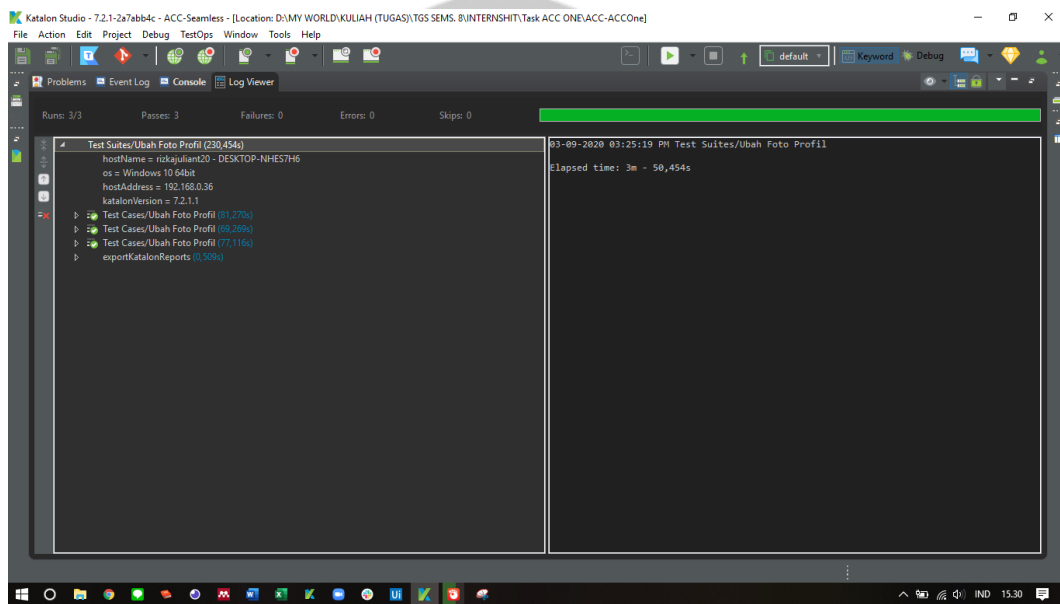
Test suite yang ditunjukkan pada Gambar 55, berfungsi untuk menjalankan *test case* mengubah foto profil yang telah dibuat sebelumnya. Pada *test suite*, variabel-variabel yang sudah diinisialisasi, dipasangkan dengan *data binding* pada Microsoft Excel, agar nilai yang dimasukkan pada saat pengujian sesuai.

	A	B	C	D	E	F	G
1	namaFoto	picture	condition				
2	gambar1.jpg	galeri	passed				
3	gambar1.jpg	kamera	passed				
4	gambar1.jpg	kamera	passedRetake				

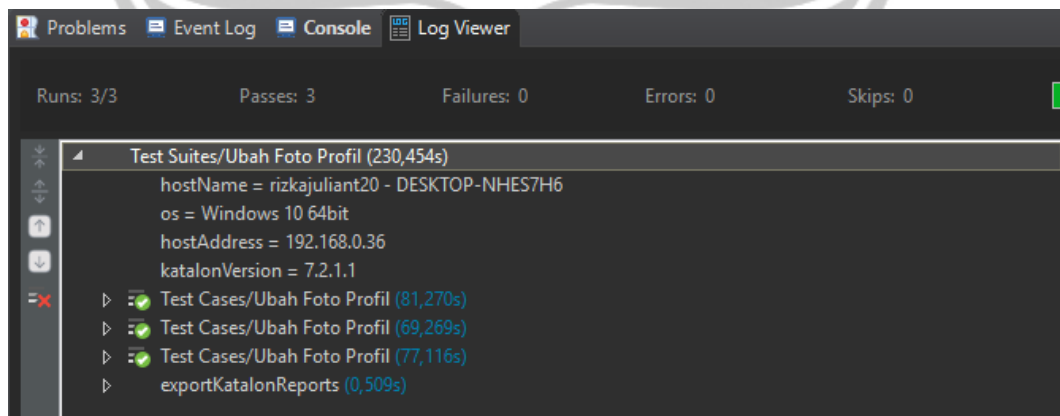
Gambar 58. *Data Binding* Untuk *Use Case* Mengubah Foto Profil.

Pada Gambar 56, ditunjukkan *data binding* yang dibuat pada Microsoft Excel, dengan nilai aktual yang disesuaikan dengan pengujian manual, di mana

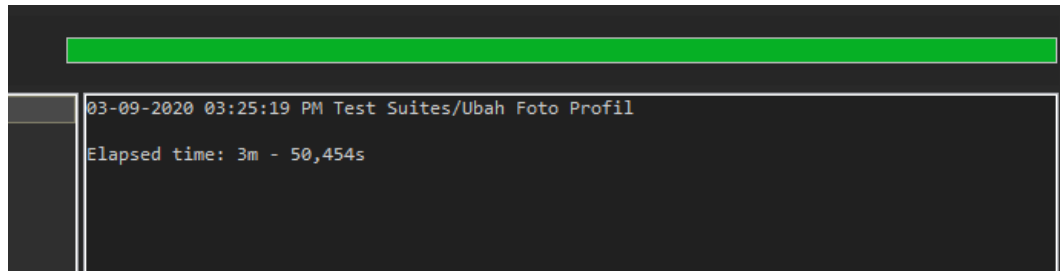
file ini akan dimasukkan ke dalam Katalon. Terdapat 3 *data binding* yang berarti akan dilakukan 3 kali pengujian dengan variasi skenario testing yang berbeda-beda untuk mendeteksi apakah terjadi *bug* dengan variasi skenario pengujian bernilai positif dan negatif. Pada *data binding* ini, terdapat kondisi passed, di mana alur yang dijalankan akan mencapai *post condition*, sedangkan kondisi passedRetake akan menjalankan skenario jika penguji ingin mengambil ulang foto melalui kamera perangkat.



Gambar 59. Hasil Automated Testing Untuk Use Case Mengubah Foto Profil (1).



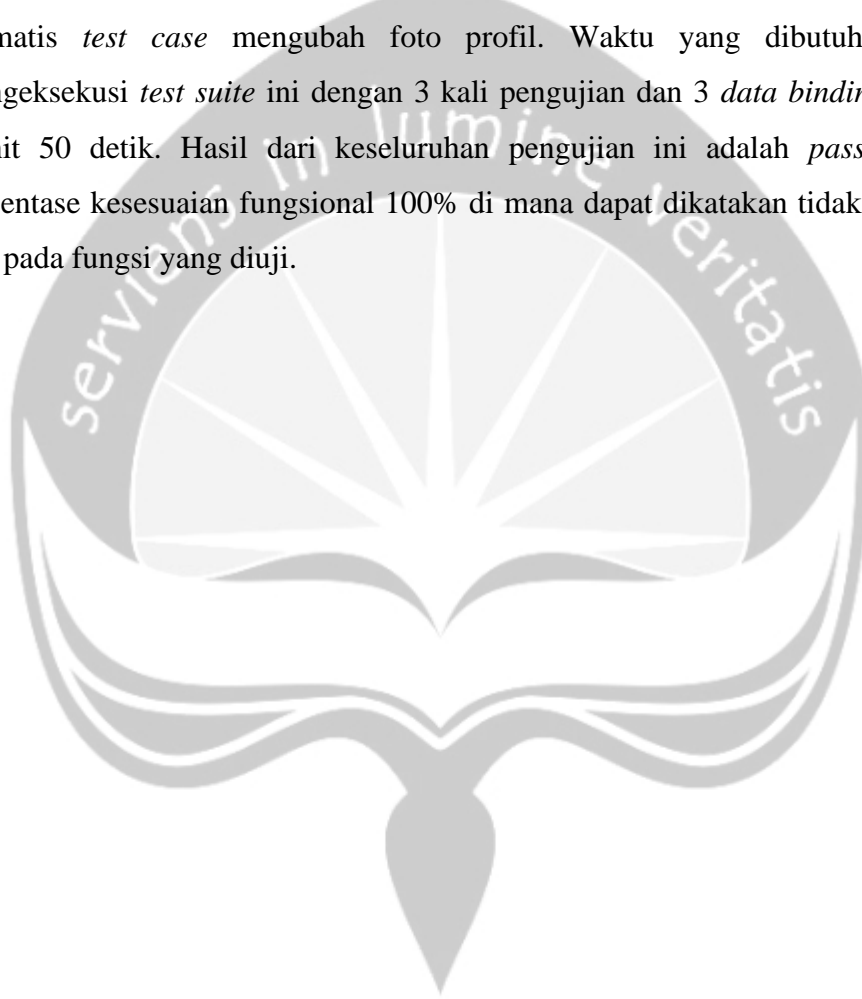
Gambar 60. Hasil Automated Testing Untuk Use Case Mengubah Foto Profil (2).



```
03-09-2020 03:25:19 PM Test Suites/Ubah Foto Profil
Elapsed time: 3m - 50,454s
```

Gambar 61. Hasil *Automated Testing* Untuk *Use Case* Mengubah Foto Profil (3).

Pada Gambar 57, 58, dan 59, ditunjukkan hasil eksekusi pengujian otomatis *test case* mengubah foto profil. Waktu yang dibutuhkan untuk mengeksekusi *test suite* ini dengan 3 kali pengujian dan 3 *data binding* adalah 3 menit 50 detik. Hasil dari keseluruhan pengujian ini adalah *passed* dengan persentase kesesuaian fungsional 100% di mana dapat dikatakan tidak ditemukan *bug* pada fungsi yang diuji.



5.2.2.6 Penerapan *Automated Testing* Pada Katalon Untuk *Use Case* Mencari Mobil Rental

Item	Object	Input	Output
1 - Start Application		"D:\MY WORLD\KULIAH (TUGAS)\TGS SEMS. 8\INTERNSHIP\Ta	
2 - Delay		10	
3 - Tap	boxCariMobil	0	
4 - Tap	boxRentalMobil	0	
5 - If Statement		newOrUsed == "used"	
5.1 - Tap	btnUsedCar	0	
6 - Tap	b3	0	
7 - Hide Keyboard			
8 - Scroll To Text		brand	
9 - Tap	tapObject	0	
10 - Delay		3	
11 - Hide Keyboard			
12 - Scroll To Text		typeMobil	
13 - If Statement		newOrUsed == "used"	
13.1 - Tap At Position		350; 766	
14 - Else Statement			
14.1 - Tap	tapObject	0	
15 - Delay		3	
16 - Hide Keyboard			
17 - If Statement		condition == "passed"	
17.1 - Scroll To Text		model	
17.2 - Tap	tapObject	0	
17.3 - Delay		3	
17.4 - Hide Keyboard			
17.5 - Scroll To Text		tahun	
17.6 - Tap	tapObject	0	

Gambar 62. *Test Case* Katalon Untuk *Use Case* Mencari Mobil Rental (1).

Item	Object	Input	Output
17.6 - Tap	tapObject	0	
17.7 - Hide Keyboard			
17.8 - Delay		3	
17.9 - ScrollUpDown.UpDown		480; 1050; 480; 230	
17.10 - Tap At Position		297; 428	
17.11 - Delay		3	
17.12 - Hide Keyboard			
17.13 - Delay		3	
17.14 - Scroll To Text		cabangACC	
17.15 - Tap	tapObject	0	
17.16 - Delay		3	
17.17 - If Statement		tujuan == "Usaha"	
17.17.1 - Tap	btnTujuanUsaha	0	
17.18 - Hide Keyboard			
17.19 - If Statement		bidangUsaha == "Pertanian"	
17.19.1 - Tap	btnPertanian	0	
17.20 - Else If Statement		bidangUsaha == "Finance"	
17.20.1 - Delay		3	
17.20.2 - Tap At Position		293; 600	
17.21 - If Statement		typeCustomer != "Personal"	
17.21.1 - Tap	android.widget.Button0 - Company	0	
17.22 - If Statement		warnaSemua == "Yes"	
17.22.1 - Tap At Position		89; 904	
17.22.2 - Tap	semuaBtn	0	
17.23 - Else Statement			
17.23.1 - If Statement		warnaHitam == "Yes"	

Gambar 63. *Test Case* Katalon Untuk *Use Case* Mencari Mobil Rental (2).

Item	Object	Input	Output
17.23 - Else Statement			
17.23.1 - If Statement		warnaHitam == "Yes"	
17.23.1.1 - Tap At Position		170; 907	
17.23.1.2 - Tap	Whitam	0	
17.23.2 - If Statement		warnaPutih == "Yes"	
17.23.2.1 - Tap At Position		243; 907	
17.23.2.2 - Tap	Wputih	0	
17.23.3 - If Statement		warnaSilver == "Yes"	
17.23.3.1 - Tap At Position		309; 907	
17.23.3.2 - Tap	Wsilver	0	
17.23.4 - If Statement		warnaMerah == "Yes"	
17.23.4.1 - Tap At Position		383; 907	
17.23.4.2 - Tap	Wmerah	0	
17.24 - Delay		5	
17.25 - Switch Statement		periodeLeasing	
17.25.1 - Case Statement		case "12":	findTestObject("Cari Mobil/Rental Mobil/line1"); 0; FailureHa
17.25.1.1 - mobile.mobileSlide.Slide			
17.25.1.2 - Break Statement			
17.25.2 - Case Statement		case "24":	findTestObject("Cari Mobil/Rental Mobil/line1"); 0.25; FailureHa
17.25.2.1 - mobile.mobileSlide.Slide			
17.25.2.2 - Break Statement			
17.25.3 - Case Statement		case "36":	findTestObject("Cari Mobil/Rental Mobil/line1"); 0.5; FailureHa
17.25.3.1 - mobile.mobileSlide.Slide			
17.25.3.2 - Break Statement			
17.25.4 - Case Statement		case "48":	findTestObject("Cari Mobil/Rental Mobil/line1"); 0.75; FailureHa
17.25.4.1 - mobile.mobileSlide.Slide			

Gambar 64. Test Case Katalon Untuk Use Case Mencari Mobil Rental (3).

Item	Object	Input	Output
17.25.4.1 - mobile.mobileSlide.Slide		findTestObject("Cari Mobil/Rental Mobil/line1"); 0.75; FailureHa	
17.25.4.2 - Break Statement			
17.25.5 - Case Statement		case "60":	findTestObject("Cari Mobil/Rental Mobil/line1"); 1; FailureHa
17.25.5.1 - mobile.mobileSlide.Slide			
17.25.5.2 - Break Statement			
17.26 - Delay		20	
17.27 - Tap	btnAjukan	0	
17.28 - Verify Element Visible	alertSedangDiproses	0	
17.29 - Tap	btnOkay	0	
18 - Else If Statement		condition == "bidangUsahalsEmptyUsaha"	
18.1 - Scroll To Text		model	
18.2 - Tap	tapObject	0	
18.3 - Delay		3	
18.4 - Hide Keyboard			
18.5 - Scroll To Text		tahun	
18.6 - Tap	tapObject	0	
18.7 - Hide Keyboard			
18.8 - Delay		3	
18.9 - ScrollUpDown.UpDown		480; 1050; 480; 230	
18.10 - Tap At Position		297; 428	
18.11 - Delay		3	
18.12 - Hide Keyboard			
18.13 - Delay		3	
18.14 - Scroll To Text		cabangACC	
18.15 - Tap	tapObject	0	
18.16 - Delay		3	

Gambar 65. Test Case Katalon Untuk Use Case Mencari Mobil Rental (4).

Item	Object	Input	Output
→x 18.15 - Tap	tapObject	0	
→x 18.16 - Delay		3	
▶ 18.17 - If Statement		tujuan == "Usaha"	
→x 18.17.1 - Tap	btnTujuanUsaha	0	
→x 18.18 - Hide Keyboard			
▶ 18.19 - If Statement		typeCustomer != "Personal"	
→x 18.19.1 - Tap	android.widget.Button0 - Company	0	
→x 18.20 - Verify Element Visible	alertHarusDiisi	0	
▶ 19 - Else If Statement		condition == "bidangUsahaDanVariabelLainIsiEmptyUsaha"	
→x 19.1 - Scroll To Text		model	
→x 19.2 - Tap	tapObject	0	
→x 19.3 - Delay		3	
→x 19.4 - Hide Keyboard			
→x 19.5 - Scroll To Text		tahun	
→x 19.6 - Tap	tapObject	0	
→x 19.7 - Hide Keyboard			
→x 19.8 - Delay		3	
→x 19.9 - ScrollUpDown.UpDown		480; 1050; 480; 230	
→x 19.10 - Tap At Position		297; 428	
→x 19.11 - Delay		3	
→x 19.12 - Hide Keyboard			
→x 19.13 - Tap At Position		680; 390	
▶ 19.14 - If Statement		tujuan == "Usaha"	
→x 19.14.1 - Tap	btnTujuanUsaha	0	
→x 19.15 - Hide Keyboard			
▶ 19.16 - If Statement		typeCustomer != "Personal"	

Gambar 66. Test Case Katalon Untuk Use Case Mencari Mobil Rental (5).

Item	Object	Input	Output
→x 19.16.1 - Tap	android.widget.Button0 - Company	0	
→x 19.17 - Verify Element Visible	alertHarusdipilih (1)	0	
→x 19.18 - Verify Element Visible	alertHarusDiisi	0	
▶ 20 - Else If Statement		condition == "variabelLainIsiEmptyUsaha"	
→x 20.1 - Scroll To Text		model	
→x 20.2 - Tap	tapObject	0	
→x 20.3 - Delay		3	
→x 20.4 - Hide Keyboard			
→x 20.5 - Scroll To Text		tahun	
→x 20.6 - Tap	tapObject	0	
→x 20.7 - Hide Keyboard			
→x 20.8 - Delay		3	
→x 20.9 - ScrollUpDown.UpDown		480; 1050; 480; 230	
→x 20.10 - Tap At Position		297; 428	
→x 20.11 - Delay		3	
→x 20.12 - Hide Keyboard			
→x 20.13 - Tap At Position		680; 390	
▶ 20.14 - If Statement		tujuan == "Usaha"	
→x 20.14.1 - Tap	btnTujuanUsaha	0	
→x 20.15 - Hide Keyboard			
▶ 20.16 - If Statement		bidangUsaha == "Pertanian"	
→x 20.16.1 - Tap	btnPertanian	0	
▶ 20.17 - Else If Statement		bidangUsaha == "Finance"	
→x 20.17.1 - Delay		3	
→x 20.17.2 - Tap At Position		293; 600	
→x 20.18 - Verify Element Visible	alertHarusdipilih (1)	0	

Gambar 67. Test Case Katalon Untuk Use Case Mencari Mobil Rental (6).

Item	Object	Input	Output
→ 20.12 - Hide Keyboard			
→ 20.13 - Tap At Position		680; 390	
▶ 20.14 - If Statement		tujuan == "Usaha"	
→ 20.14.1 - Tap	btnTujuanUsaha	0	
→ 20.15 - Hide Keyboard			
▶ 20.16 - If Statement		bidangUsaha == "Pertanian"	
→ 20.16.1 - Tap	btnPertanian	0	
▶ 20.17 - Else If Statement		bidangUsaha == "Finance"	
→ 20.17.1 - Delay		3	
→ 20.17.2 - Tap At Position		293; 600	
→ 20.18 - Verify Element Visible	alertHarusdipilih (1)	0	
▶ 21 - Else Statement			
→ 21.1 - Scroll To Text		model	
→ 21.2 - Tap	tapObject	0	
→ 21.3 - Delay		3	
→ 21.4 - Hide Keyboard			
→ 21.5 - Scroll To Text		tahun	
→ 21.6 - Tap	tapObject	0	
→ 21.7 - Hide Keyboard			
→ 21.8 - Delay		3	
→ 21.9 - ScrollUpDown.UpDown		480; 1050; 480; 230	
→ 21.10 - Tap At Position		297; 428	
→ 21.11 - Delay		3	
→ 21.12 - Hide Keyboard			
→ 21.13 - Tap At Position		680; 390	
→ 21.14 - Verify Element Visible	alertHarusdipilih (1)	0	

Gambar 68. *Test Case* Katalon Untuk *Use Case* Mencari Mobil Rental (7).

Pada Gambar 60 hingga 66, merupakan rancangan *test case* untuk menguji fungsi mencari mobil rental. Fungsi ini diuji pada aplikasi acc.one dengan versi *update* per tanggal 21 Januari 2020. Pada dasarnya, fungsi mencari promo mobil tertentu ini tergolong ke dalam *use case* melakukan rental mobil *online* yang dijabarkan pada Gambar 3. Alur pengujian yang diterapkan untuk *automated testing* ini sesuai dengan alur pengujian manual. Pada *test case* ini terdapat instruksi *Tap At Position* yang berfungsi untuk *tap* beberapa objek yang tidak dapat dideteksi seperti halnya pada poin 20.10 yang merupakan area pengajuan, dan poin 20.17.2 yang merupakan *drop down* dari bidang usaha yang bernilai *finance*. Instruksi *Verify Element Visible* pada kasus ini digunakan untuk mengecek adanya objek peringatan bahwa salah satu variabel harus dipilih dan objek yang menandakan bahwa pengajuan mobil rental sedang diproses. Pembuatan objek baru pada *test case* ini digunakan untuk memilih beberapa objek *list* dari *drop down* sesuai dengan nilai aktual variabel tertentu. Pengaturan pembuatan objek baru sama dengan Gambar 19 dan 20 dengan variabel yang disesuaikan dengan kebutuhan.

No.	Name	Type	Default value
1	newOrUsed	String	""
2	brand	String	""
3	typeMobil	String	""
4	model	String	""
5	tahun	String	""
6	areaPengajuan	String	""
7	cabangACC	String	""
8	tujuan	String	""
9	bidangUsaha	String	""
10	typeCustomer	String	""
11	warnaHitam	String	""
12	warnaPutih	String	""
13	warnaSilver	String	""
14	warnaMerah	String	""
15	warnaSemua	String	""
16	periodeLeasing	String	""
17	condition	String	""

Gambar 69. Variabel Katalon Untuk *Use Case* Mencari Mobil Rental.

Pada Gambar 67, ditunjukkan variabel-variabel yang digunakan pada pengujian otomatis di Katalon untuk *use case* mencari mobil rental. Variabel condition digunakan sebagai alat untuk membantu menentukan alur alternatif, sehingga alur pengujian dapat lebih singkat. Variabel condition juga dapat digunakan untuk mengurangi penggunaan *if-else* di dalam *if-else*. Dengan membuat kondisi yang lebih bervariasi, pembuatan *test case* nantinya akan lebih singkat dan mudah dipahami dengan mengandalkan nilai dari variabel condition sebagai kondisi di *if-else* dalam penentuan alur pengujian.

No.	ID	Data Iteration	Type
1	Data Files/Car/Mobil/cariMobilRent	All	One

No.	Name	Default value	Type	Test Data
1	newOrUsed	""	Data Column	1 - Data Files/Car
2	brand	""	Data Column	1 - Data Files/Car
3	typeMobil	""	Data Column	1 - Data Files/Car
4	model	""	Data Column	1 - Data Files/Car
5	tahun	""	Data Column	1 - Data Files/Car
6	areaPengajuan	""	Data Column	1 - Data Files/Car
7	cabangACC	""	Data Column	1 - Data Files/Car
8	tujuan	""	Data Column	1 - Data Files/Car
9	bidangUsaha	""	Data Column	1 - Data Files/Car
10	typeCustomer	""	Data Column	1 - Data Files/Car
11	warnaHitam	""	Data Column	1 - Data Files/Car

Gambar 70. *Test Suite* Katalon Untuk *Use Case* Mencari Mobil Rental.

Test suite yang ditunjukkan pada Gambar 68, berfungsi untuk menjalankan *test case* mencari mobil rental yang telah dibuat sebelumnya. Pada *test suite*, variabel-variabel yang sudah diinisialisasi, dipasangkan dengan variabel *data binding* pada Microsoft Excel, agar nilai yang dimasukkan pada saat pengujian sesuai.

	A	B	C	D	E	F	G	H
1	newOrUsed	brand	typeMobil	model	tahun	areaPengajuan	cabangACC	tujuan k
2	new	TOYOTA	ALL NEW AVANZA	1.3 E A/T	2019	DKI Jakarta, Banten, Jawa Barat	KARAWACI	Pribadi
3	used	DAIHATSU	AYLA	T:1.0 D+M/T NEW	2018	DKI Jakarta, Banten, Jawa Barat	PLUIT	Usaha
4	used	DAIHATSU	AYLA	T:1.0 D+M/T NEW	2018	DKI Jakarta, Banten, Jawa Barat	PLUIT	Usaha
5	used	DAIHATSU	AYLA	T:1.0 D+M/T NEW	2018	DKI Jakarta, Banten, Jawa Barat		Usaha
6	used	DAIHATSU	AYLA	T:1.0 D+M/T NEW	2018	DKI Jakarta, Banten, Jawa Barat		Usaha
7	new	TOYOTA	ALL NEW AVANZA	1.3 E A/T	2019	DKI Jakarta, Banten, Jawa Barat		Pribadi
8								

Gambar 71. *Data Binding* Untuk *Use Case* Mencari Mobil Rental (1).

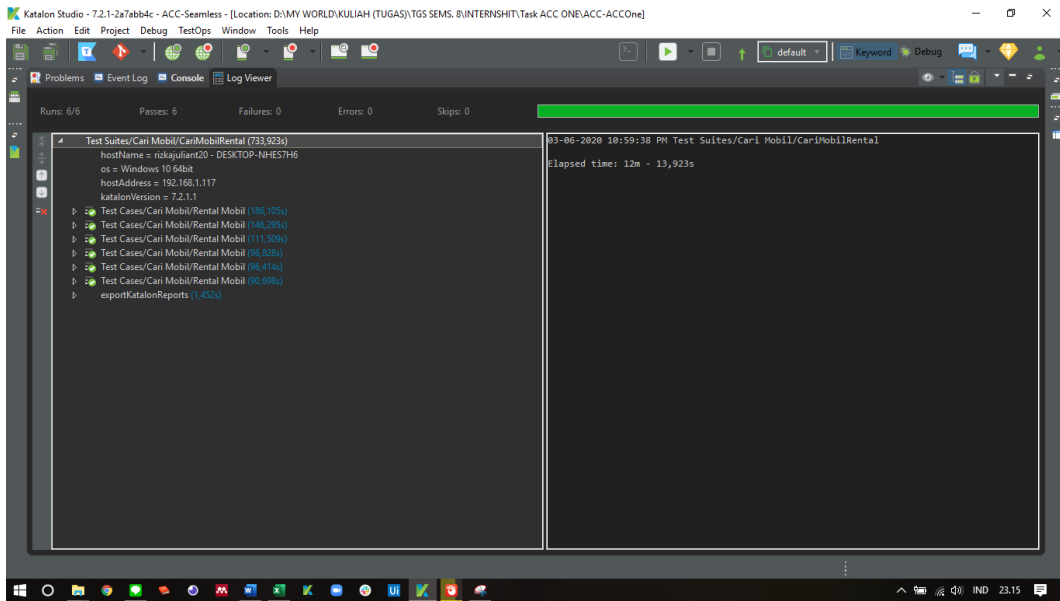
H	I	J	K	L	M	N	O	P
tujuan	bidangUsaha	typeCustomer	warnaHitam	warnaPutih	warnaSilver	warnaMerah	warnaSemua	periodeLeasing
Pribadi		Personal	Yes	Yes				48
Usaha	Pertanian	Company			Yes	Yes		36
Usaha	Finance	Company		Yes	Yes	Yes		36
Usaha	Pertanian	Company					Yes	24
Usaha	Pertanian	Company					Yes	24
Pribadi		Personal		Yes	Yes			60

Gambar 72. *Data Binding* Untuk *Use Case* Mencari Mobil Rental (2).

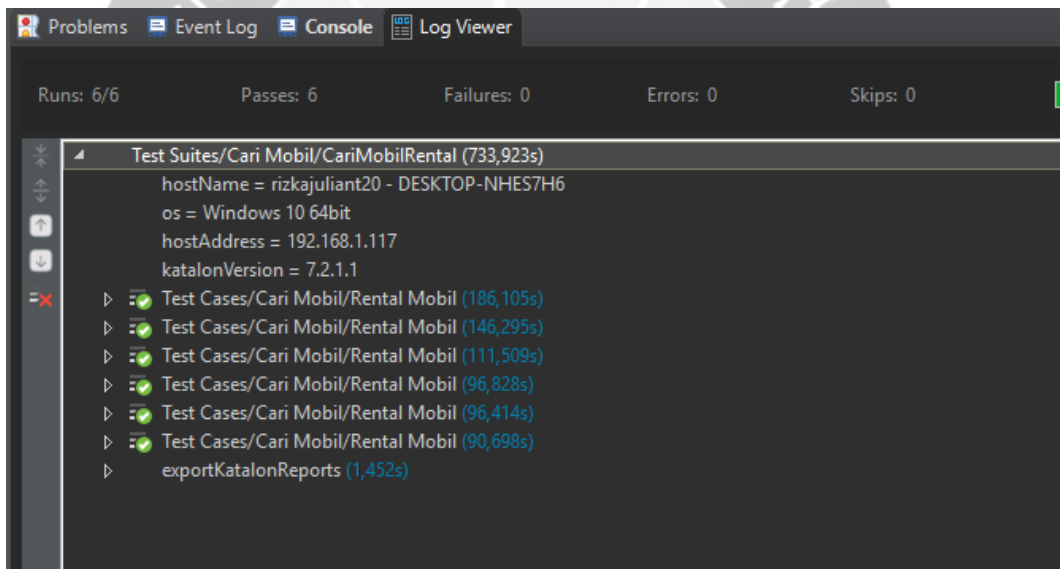
L	M	N	O	P	Q
warnaPutih	warnaSilver	warnaMerah	warnaSemua	periodeLeasing	condition
Yes				48	passed
	Yes	Yes		36	passed
Yes	Yes	Yes		36	bidangUsahalsEmptyUsaha
			Yes	24	bidangUsahaDanVariabelLainIsEmptyUsaha
			Yes	24	variabelLainIsEmptyUsaha
Yes	Yes			60	variabelLainIsEmpty

Gambar 73. *Data Binding* Untuk *Use Case* Mencari Mobil Rental (3).

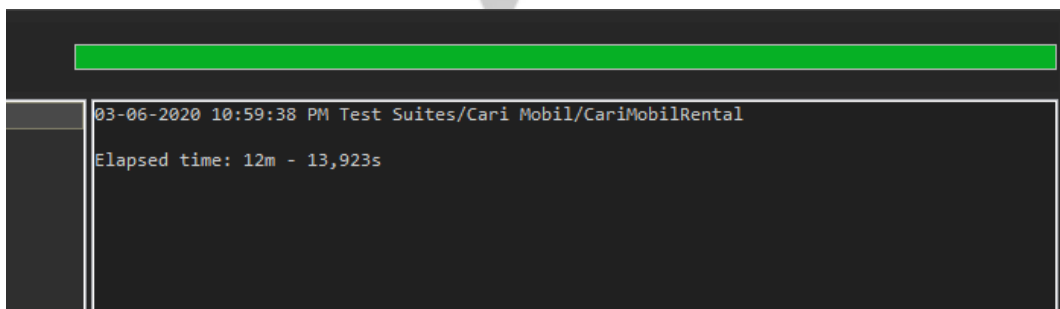
Pada Gambar 69, 70, dan 71, ditunjukkan *data binding* yang dibuat pada Microsoft Excel, dengan nilai aktual yang disesuaikan dengan pengujian manual, di mana *file* ini akan dimasukkan ke dalam Katalon. Terdapat 6 *data binding* yang berarti akan dilakukan 6 kali pengujian dengan variasi skenario *testing* yang berbeda-beda untuk mendeteksi apakah terjadi *bug* dengan variasi skenario pengujian bernilai positif dan negatif. Pada *data binding* ini, terdapat kondisi *passed*, di mana alur yang dijalankan akan mencapai *post condition*. Untuk kondisi lainnya merupakan alur ketika terdapat 1 atau beberapa variabel tidak dipilih di mana hal ini dijelaskan pada *alternate flow* di Bab 4.



Gambar 74. Hasil *Automated Testing* Untuk *Use Case* Mencari Mobil Rental (1).



Gambar 75. Hasil *Automated Testing* Untuk *Use Case* Mencari Mobil Rental (2).



Gambar 76. Hasil *Automated Testing* Untuk *Use Case* Mencari Mobil Rental (3).

Pada Gambar 72, 73, dan 74, ditunjukkan hasil eksekusi pengujian otomatis *test case* mencari mobil rental. Waktu yang dibutuhkan untuk mengeksekusi test suite ini dengan 6 kali pengujian dan 6 *data binding* adalah 12 menit 13 detik. Hasil dari keseluruhan pengujian ini adalah *passed* dengan persentase kesesuaian fungsional 100% di mana dapat dikatakan tidak ditemukan *bug* pada fungsi yang diuji.



5.2.2.7 Penerapan *Automated Testing* Pada Katalon Untuk *Use Case* Lupa Kata Sandi

Item	Object	Input
1 - Start Application		"D:\MY WORLD\KULIAH (TUGAS)\TGS SEMS. 8\INTERNSHIP\Backup\ACC-ACCONElac
2 - Delay		20
3 - Tap At Position		443; 229
4 - Delay		5
5 - Tap	btnLupaKataSandi	0
6 - Delay		5
7 - Tap	txtEmail	0
8 - Set Text	editTextEmail	email; 0
9 - Tap	btnKonfirmasi	0
10 - If Statement		condition == "emailSalah"
10.1 - Verify Element Visible	alertEmailSalah	0
11 - Else Statement		5
11.1 - Delay		5
11.2 - If Statement		kirimOTPke == "email"
11.2.1 - Tap	kirimKeEmail	0
11.3 - Else Statement		0
11.3.1 - Tap	kirimSMS	0
11.4 - If Statement		Mobile.verifyElementVisible(findTestObject("Lupa Kata Sandi/checkBoxCaptcha"), 0, Fa
11.4.1 - Tap	checkBoxCaptcha	0
11.4.2 - Delay		30
11.4.3 - Tap	SubmitCaptcha	0
11.5 - If Statement		condition == "kodeOTPSalah"
11.5.1 - Set Text	editTextOTP	OTP; 0
11.5.2 - Tap	btnVerifikasi	0
11.5.3 - Delay		5
11.5.4 - Verify Element Visible	alertKodeOTPSalah	0

Gambar 77. *Test Case* Katalon Untuk *Use Case* Lupa Kata Sandi (1).

Item	Object	Input
11.5.4 - Verify Element Visible	alertKodeOTPSalah	0
11.6 - Else Statement		50
11.6.1 - Delay		0
11.6.2 - Tap	btnVerifikasi	0
11.6.3 - Verify Element Not Visible	alertKodeOTPSalah	0
11.6.4 - Delay		5
11.6.5 - Set Text	editTextSandi1	kataSandi; 0
11.6.6 - Set Text	editTextSandi2	kataSandi2; 0
11.6.7 - If Statement		condition == "minimalAda1Angka"
11.6.7.1 - Verify Element Visible	alertFormatBelumSesuai	0
11.6.7.2 - Tap	btnResetDanMasuk	0
11.6.7.3 - Verify Element Visible	alertMin1Angka	0
11.6.8 - Else If Statement		condition == "harus7Karakter"
11.6.8.1 - Verify Element Visible	alertMin7KarBiasa	0
11.6.8.2 - Tap	btnResetDanMasuk	0
11.6.8.3 - Verify Element Visible	alertMin7KarFinal	0
11.6.9 - Else If Statement		condition == "passwordHarusDiisi"
11.6.9.1 - Verify Element Visible	alertPasswordHarusDiisi	0
11.6.9.2 - Tap	btnResetDanMasuk	0
11.6.9.3 - Verify Element Visible	alertMin7KarFinal	0
11.6.10 - Else If Statement		condition == "minimalAda1Huruf"
11.6.10.1 - Verify Element Visible	alertFormatBelumSesuai	0
11.6.10.2 - Tap	btnResetDanMasuk	0
11.6.10.3 - Verify Element Visible	alertMin1Huruf	0
11.6.11 - Else If Statement		condition == "passwordTidakSesuaiHarusDiisi"
11.6.11.1 - Tap	btnResetDanMasuk	0

Gambar 78. *Test Case* Katalon Untuk *Use Case* Lupa Kata Sandi (2).

Item	Object	Input
→ 11.6.7.3 - Verify Element Visible	alertMin1Angka	0
▶ 11.6.8 - Else If Statement		condition == "harus7Karakter"
→ 11.6.8.1 - Verify Element Visible	alertMin7KarBiasa	0
→ 11.6.8.2 - Tap	btnResetDanMasuk	0
→ 11.6.8.3 - Verify Element Visible	alertMin7KarFinal	0
▶ 11.6.9 - Else If Statement		condition == "passwordHarusDiisi"
→ 11.6.9.1 - Verify Element Visible	alertPasswordHarusDiisi	0
→ 11.6.9.2 - Tap	btnResetDanMasuk	0
→ 11.6.9.3 - Verify Element Visible	alertMin7KarFinal	0
▶ 11.6.10 - Else If Statement		condition == "minimalAda1Huruf"
→ 11.6.10.1 - Verify Element Visible	alertFormatBelumSesuai	0
→ 11.6.10.2 - Tap	btnResetDanMasuk	0
→ 11.6.10.3 - Verify Element Visible	alertMin1Huruf	0
▶ 11.6.11 - Else If Statement		condition == "passwordTidakSesuaiHarusDiisi"
→ 11.6.11.1 - Tap	btnResetDanMasuk	0
→ 11.6.11.2 - Verify Element Visible	Konfirmasi Kata Sandi harus diisi	0
▶ 11.6.12 - Else If Statement		condition == "passwordTidakSesuai"
→ 11.6.12.1 - Tap	btnResetDanMasuk	0
→ 11.6.12.2 - Verify Element Visible	alertPasswordTidakSesuai	0
▶ 11.6.13 - Else If Statement		condition == "passed"
→ 11.6.13.1 - Verify Element Not Visible	alertMin1Angka	0
→ 11.6.13.2 - Verify Element Not Visible	alertMin1Huruf	0
→ 11.6.13.3 - Verify Element Not Visible	alertMin7KarBiasa	0
→ 11.6.13.4 - Verify Element Not Visible	alertMin7KarFinal	0
→ 11.6.13.5 - Tap	btnResetDanMasuk	0
→ 11.6.13.6 - Verify Element Visible	alertPasswordBerhasilDiganti	0

Gambar 79. Test Case Katalon Untuk Use Case Lupa Kata Sandi (3).

Pada Gambar 75, 76, dan 77, merupakan rancangan *test case* untuk menguji fungsi lupa kata sandi. Fungsi ini diuji pada aplikasi acc.one dengan versi *update* per tanggal 12 Februari 2020. Pada dasarnya, fungsi mencari promo mobil tertentu ini tergolong ke dalam *use case* mengelola data akun *customer* yang dijabarkan pada Gambar 3. Alur pengujian yang diterapkan untuk *automated testing* ini sesuai dengan alur pengujian manual. Pada *test case* terdapat instruksi *Tap At Position* yang berfungsi untuk *tap button* Daftar dan Masuk karena objek tersebut tidak dapat dideteksi oleh Katalon. Terdapat pula kondisi yang mengharuskan sistem mendeteksi terlebih dahulu untuk melakukan langkah selanjutnya yaitu ketika sudah melakukan pergantian kata sandi sebanyak 3 kali, maka diharuskan mengisi Captcha yang tersedia. Pada kasus tersebut objek yang dideteksi adalah sebagian dari Captcha yang muncul, dan ketika terdeteksi maka langkah yang dilakukan selanjutnya adalah mencentang "I am not a robot" dan *tap button* submit. Instruksi Verify Element Visible dan Verify Element Not Visible dalam hal ini akan mengecek adanya peringatan-peringatan yang muncul pada saat *validity check* di beberapa variabel tertentu seperti *text field* untuk kata sandi tidak boleh kosong, maka yang akan dideteksi di sini adalah objek peringatan bahwa kata sandi harus diisi. Pengisian kode OTP dilakukan secara manual karena kode-kode OTP tentunya berbeda-beda dan tidak dapat diprediksi, dan juga tidak ada kode OTP *default* yang disediakan oleh pengembang untuk dapat digunakan

dalam pengujian. Maka dari itu, pengisian kode OTP diganti dengan instruksi *delay* atau menunda waktu pengujian selama 50 detik untuk dapat diisi secara manual.

No.	Name	Type	Default value
1	email	String	""
2	kirimOTPke	String	""
3	OTP	String	""
4	kataSandi	String	""
5	kataSandi2	String	""
6	condition	String	""

Gambar 80. Variabel Katalon Untuk *Use Case* Lupa Kata Sandi.

Pada Gambar 78, ditunjukkan variabel-variabel yang digunakan pada pengujian otomatis di Katalon untuk *use case* lupa kata sandi. Variabel *condition* digunakan sebagai alat untuk membantu menentukan alur alternatif, sehingga alur pengujian dapat lebih singkat. Variabel *condition* juga dapat digunakan untuk mengurangi penggunaan *if-else* di dalam *if-else*. Dengan membuat kondisi yang lebih bervariasi, pembuatan *test case* nantinya akan lebih singkat dan mudah dipahami dengan mengandalkan nilai dari variabel *condition* sebagai kondisi di *if-else* dalam penentuan alur pengujian.

The screenshot displays the Katalon Studio interface with the following sections:

- Execution Information:** Shows a table with one test case:

No.	ID	Description	Run
1	Test Cases/Lupa Kata Sandi		<input type="checkbox"/>
- Test Data:** Shows a table with one data iteration:

No.	ID	Data Iteration	Type
1	Data Files/Lupa Kata Sandi	1-10	One
- Variable Binding:** Shows a table mapping variables to test data:

No.	Name	Default value	Type	Test Data
1	email	""	Data Column	1 - Data Files/Lupa Kata Sandi
2	kirimOTPke	""	Data Column	1 - Data Files/Lupa Kata Sandi
3	kataSandi	""	Data Column	1 - Data Files/Lupa Kata Sandi
4	kataSandi2	""	Data Column	1 - Data Files/Lupa Kata Sandi
5	condition	""	Data Column	1 - Data Files/Lupa Kata Sandi
6	OTP	""	Data Column	1 - Data Files/Lupa Kata Sandi

Gambar 81. *Test Suite* Katalon Untuk *Use Case* Lupa Kata Sandi.

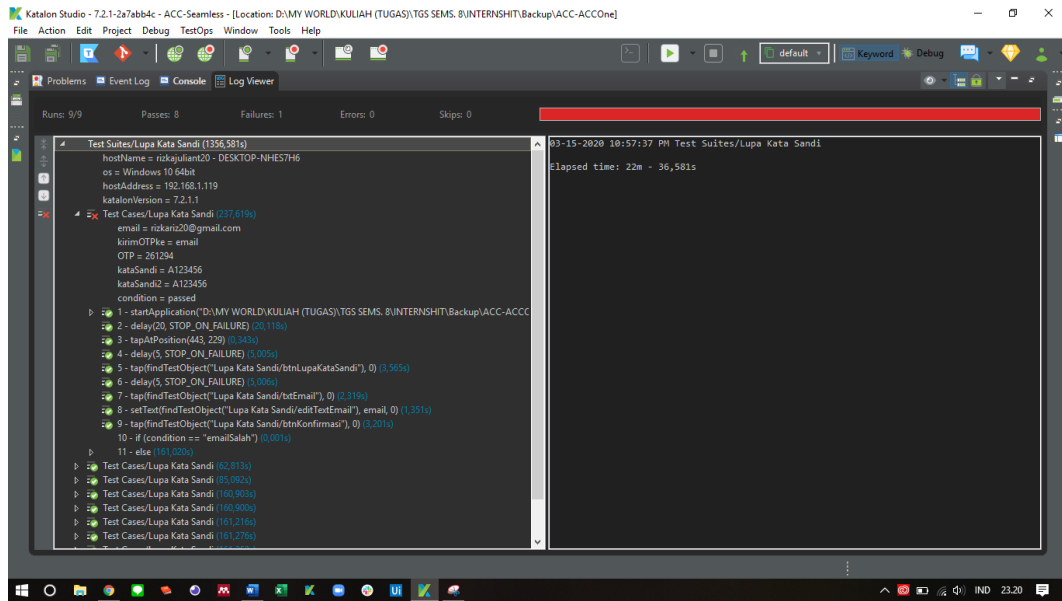
Test suite yang ditunjukkan pada Gambar 79, berfungsi untuk menjalankan *test case* lupa kata sandi yang telah dibuat sebelumnya. Pada *test*

suite, variabel-variabel yang sudah diinisialisasi, dipasangkan dengan variabel *data binding* pada Microsoft Excel, agar nilai yang dimasukkan pada saat pengujian sesuai.

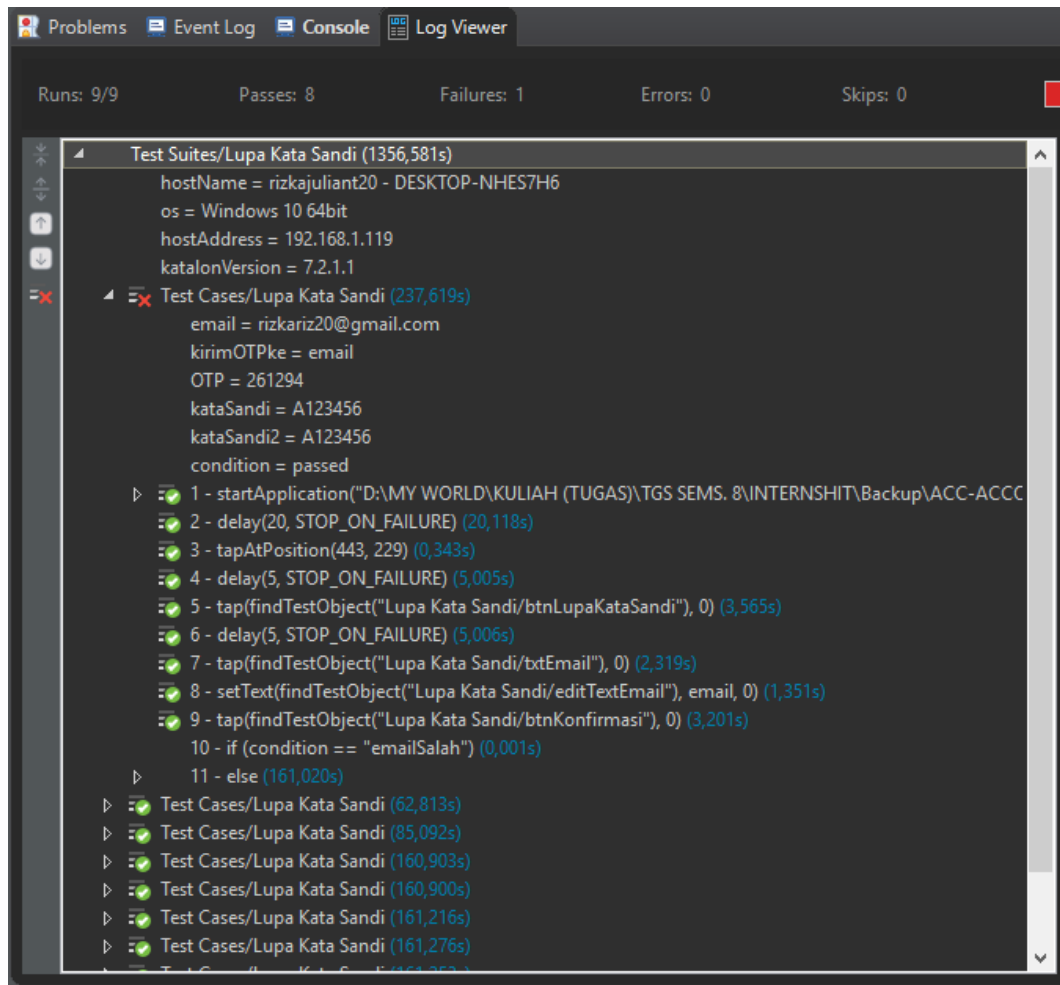
	A	B	C	D	E	F
1	email	kirimOTPke	OTP	kataSandi	kataSandi2	condition
2	rizkariz20@gmail.com	email	261294	A123456	A123456	passed
3	tralalala@gmail.com		261294			emailSalah
4	rizkariz20@gmail.com	email	444444			kodeOTPsalah
5	rizkariz20@gmail.com	email	261294			passwordHarusDiisi
6	rizkariz20@gmail.com	email	261294	IN456		harus7Karakter
7	rizkariz20@gmail.com	email	261294	12345678		minimalAda1Huruf
8	rizkariz20@gmail.com	email	261294	AAAAAAAA		minimalAda1Angka
9	rizkariz20@gmail.com	email	261294	A123456	IN1234	passwordTidakSesuai
10	rizkariz20@gmail.com	email	261294	A123456		passwordTidakSesuaiHarusDiisi

Gambar 82. *Data Binding* Untuk *Use Case* Lupa Kata Sandi.

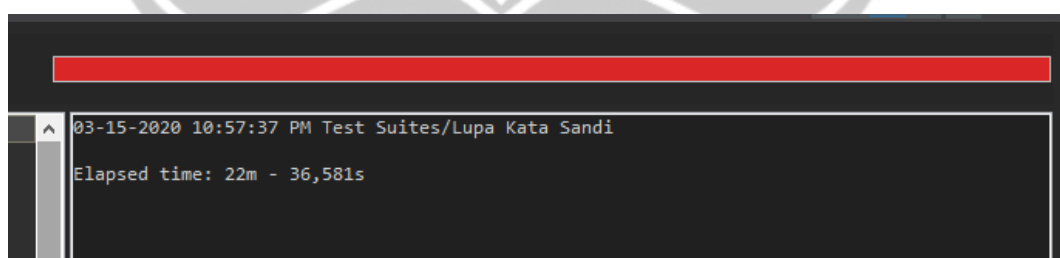
Pada Gambar 80, ditunjukkan *data binding* yang dibuat pada Microsoft Excel, dengan nilai aktual yang disesuaikan dengan pengujian manual, di mana *file* ini akan dimasukkan ke dalam Katalon. Terdapat 9 data binding yang berarti akan dilakukan 9 kali pengujian dengan variasi skenario *testing* yang berbeda-beda untuk mendeteksi apakah terjadi *bug* dengan variasi skenario pengujian bernilai positif dan negatif. Pada *data binding* ini, terdapat kondisi *passed*, di mana alur yang dijalankan akan mencapai *post condition*, kondisi *emailSalah* ketika email yang dimasukkan tidak sesuai dengan format atau salah, kondisi *kodeOTPsalah* adalah ketika kode OTP yang dimasukkan tidak sesuai dengan yang dikirimkan melalui alamat email atau nomor telepon, kondisi *passwordHarusDiisi* adalah ketika *text field* kata sandi tidak diisi, kondisi *harus7Karakter* adalah ketika kata sandi yang dimasukkan kurang dari 7 karakter alfanumerik, kondisi *minimalAda1Huruf* adalah ketika masukan hanya berupa angka, kondisi *minimalAda1Angka* adalah ketika masukan hanya berupa huruf, dan kondisi *passwordTidakSesuai* adalah ketika *text field* konfirmasi kata sandi kosong atau tidak sesuai dengan kata sandi baru.



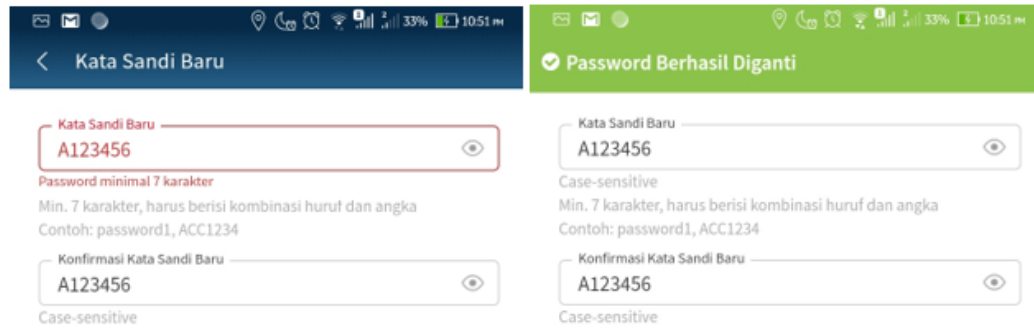
Gambar 83. Hasil Automated Testing Untuk Use Case Lupa Kata Sandi (1).



Gambar 84. Hasil *Automated Testing* Untuk *Use Case* Lupa Kata Sandi (2).



Gambar 85. Hasil *Automated Testing* Untuk *Use Case* Lupa Kata Sandi (3).



Gambar 86. *Bug* yang Ditemukan Pada *Use Case* Lupa Kata Sandi.

Pada Gambar 81, 82, dan 83, ditunjukkan hasil eksekusi pengujian otomatis *test case* lupa kata sandi. Waktu yang dibutuhkan untuk mengeksekusi *test suite* ini dengan 9 kali pengujian dan 9 *data binding* adalah 22 menit 36 detik. Hasil dari keseluruhan pengujian ini adalah *passed* dengan persentase kesesuaian fungsional 88,89% di mana 8 dari 9 *data binding* berhasil dijalankan sedangkan 1 *data binding* gagal. *Bug* ditunjukkan pada Gambar 84, di mana *text field* kata sandi baru yang mengeluarkan peringatan 'kata sandi minimal 7 karakter', sedangkan masukan sudah berjumlah 7 karakter. Hal tersebut menunjukkan kesesuaian hasil yang dikeluarkan dari pengujian otomatis dan pengujian manual.

5.2.2.8 Penerapan *Automated Testing* Pada Katalon Untuk *Use Case* Melihat Status Pengajuan Mobil

Item	Object	Input
1 - Start Application		"D:\MY WORLD\KULIAH (TUGAS)\TGS SEMS. 8\INTERNSHIP\Task ACC ONE\ACC-ACCOne\acc.one (2).s
2 - Delay		20
3 - Tap	btnStatusPeng	0
4 - If Statement		condition == "searchByText"
4.1 - Set Text	editTextCariMobilPengajuan	namaMobil; 0
4.2 - Delay		10
4.3 - Tap	tapObject	0
4.4 - Delay		10
4.5 - Verify Element Visible	btnLihat Detail Pengajuan	0
4.6 - Tap	btnLihat Detail Pengajuan	0
4.7 - Delay		10
4.8 - Delay		5
5 - Else If Statement		condition == "searchByScroll"
5.1 - Scroll To Text		namaMobil
5.2 - Delay		10
5.3 - Tap	tapObject	0
5.4 - Delay		10
5.5 - Verify Element Visible	btnLihat Detail Pengajuan	0
5.6 - Tap	btnLihat Detail Pengajuan	0
5.7 - Delay		10
5.8 - Delay		5
6 - Else Statement		
6.1 - Set Text	editTextCariMobilPengajuan	namaMobil; 0
6.2 - Delay		10
6.3 - Verify Element Not Visible	tapObject	0
7 - Press Back		

Gambar 87. *Test Case* Katalon Untuk *Use Case* Melihat Status Pengajuan Mobil.

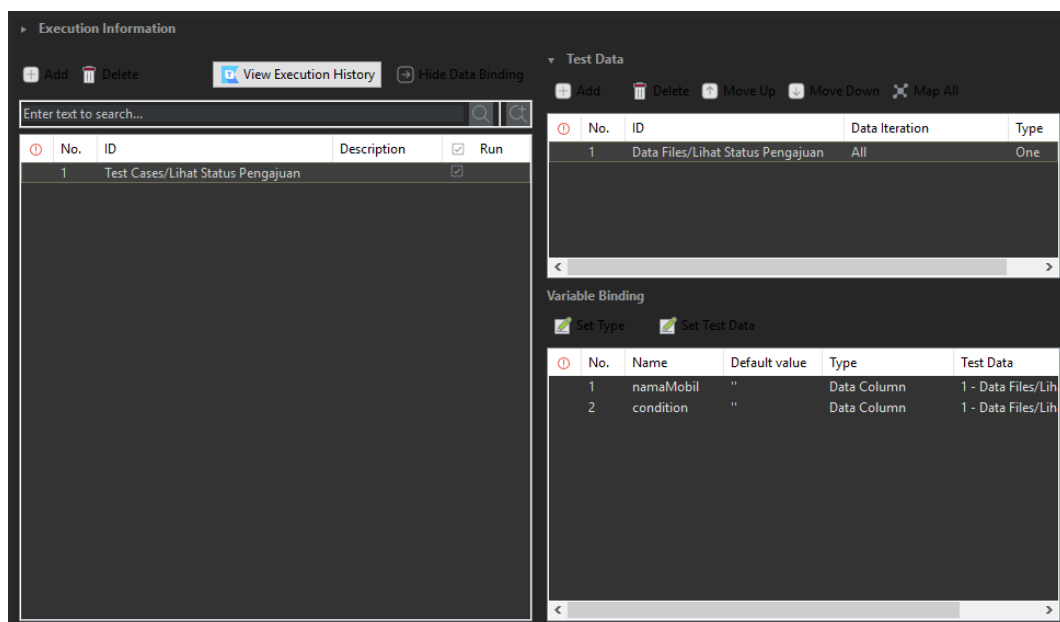
Pada Gambar 85, merupakan rancangan *test case* untuk menguji fungsi melihat status pengajuan mobil. Fungsi ini diuji pada aplikasi acc.one dengan versi *update* per tanggal 12 Februari 2020. Pada dasarnya, fungsi melihat status pengajuan mobil ini tergolong ke dalam *use case* mengelola pembelian mobil *online* yang dijabarkan pada Gambar 3. Alur pengujian yang diterapkan untuk *automated testing* ini sesuai dengan alur pengujian manual. Pada *test case* pembuatan objek baru yang berfungsi untuk mencari objek berdasarkan teks dari nilai variabel namaMobil. Instruksi *Verify Element Visible* pada *test case* ini berfungsi untuk mendeteksi tidak adanya objek peringatan tidak ditemukannya mobil, seperti yang ditunjukkan pada Gambar 85 poin 6.3.

No.	Name	Type	Default value
1	namaMobil	String	""
2	condition	String	""

Gambar 88. Variabel Katalon Untuk *Use Case* Melihat Status Pengajuan Mobil.

Pada Gambar 86, ditunjukkan variabel-variabel yang digunakan pada pengujian otomatis di Katalon untuk *use case* melihat status pengajuan mobil. Variabel condition digunakan sebagai alat untuk membantu menentukan alur

alternatif, sehingga alur pengujian dapat lebih singkat. Variabel condition juga dapat digunakan untuk mengurangi penggunaan *if-else* di dalam *if-else*. Dengan membuat kondisi yang lebih bervariasi, pembuatan *test case* nantinya akan lebih singkat dan mudah dipahami dengan mengandalkan nilai dari variabel condition sebagai kondisi di *if-else* dalam penentuan alur pengujian.



Gambar 89. *Test Suite* Katalon Untuk *Use Case* Melihat Status Pengajuan Mobil.

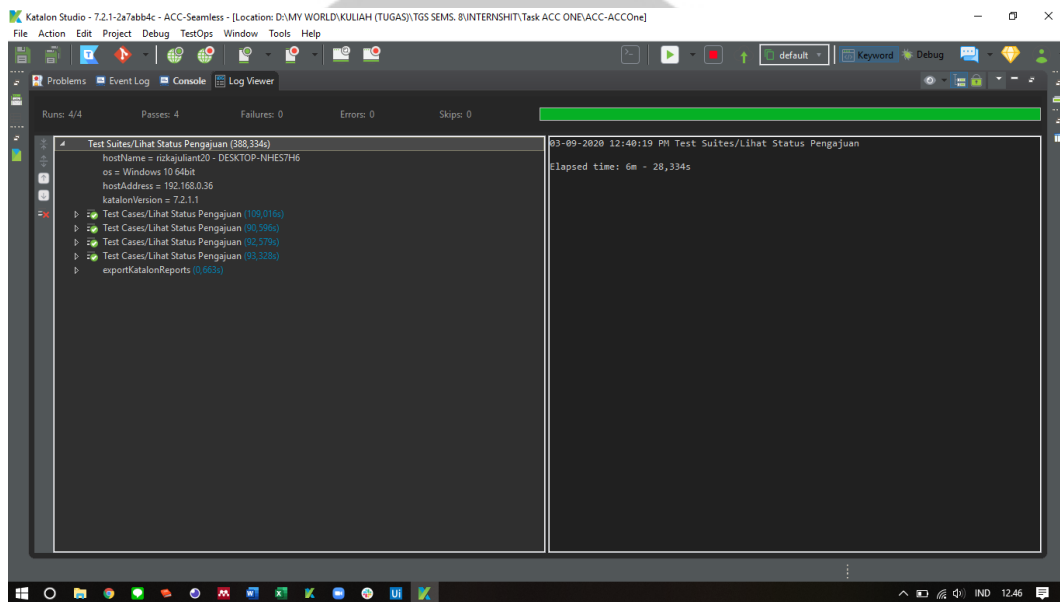
Test suite yang ditunjukkan pada Gambar 87, berfungsi untuk menjalankan *test case* melihat status pengajuan mobil yang telah dibuat sebelumnya. Pada *test suite*, variabel-variabel yang sudah diinisialisasi, dipasangkan dengan variabel *data binding* pada Microsoft Excel, agar nilai yang dimasukkan pada saat pengujian sesuai.

	A	B	C	D	E	F	G	H	I
1	namaMobil	condition							
2	Agya	searchByText							
3	Calya	searchByScroll							
4	Daihatsu New Ayla	searchByText							
5	Brio	failed							

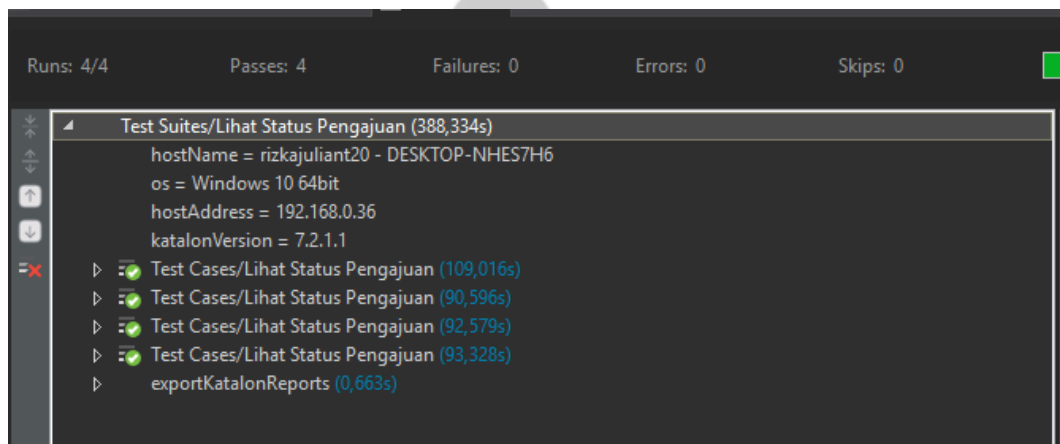
Gambar 90. *Data Binding* Untuk *Use Case* Melihat Status Pengajuan Mobil.

Pada Gambar 88, ditunjukkan *data binding* yang dibuat pada Microsoft Excel, dengan nilai aktual yang disesuaikan dengan pengujian manual, di mana *file* ini akan dimasukkan ke dalam Katalon. Terdapat 4 *data binding* yang berarti

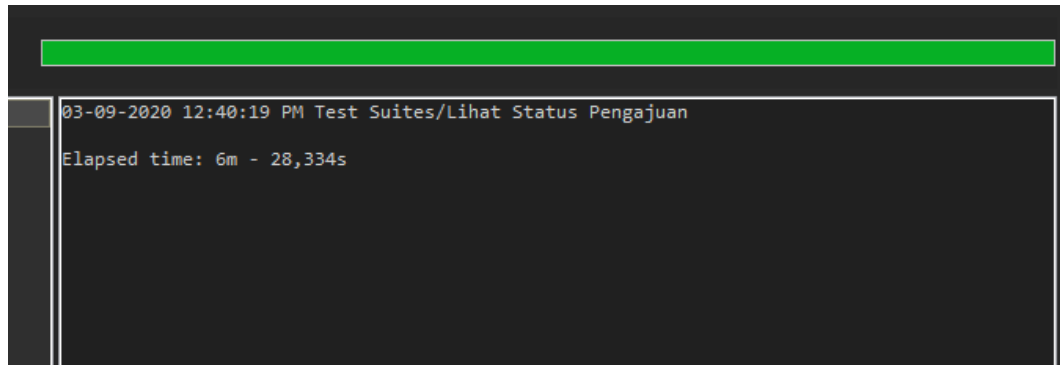
akan dilakukan 4 kali pengujian dengan variasi skenario *testing* yang berbeda-beda untuk mendeteksi apakah terjadi *bug* dengan variasi skenario pengujian bernilai positif dan negatif. Pada *data binding* ini, terdapat kondisi *searchByText* dan *searchByScroll* di mana alur yang dijalankan akan mencapai *post condition* namun berbeda alur, jika *searchByText* akan mencari mobil dengan menuliskan nama mobil pada *text field* pencarian, sedangkan *searchByScroll* akan mencari dengan *scroll* hingga ketemu dengan nama mobil yang diinginkan. Untuk kondisi *failed* alur yang dijalankan tidak akan mencapai *post condition*.



Gambar 91. Hasil *Automated Testing* Untuk *Use Case* Melihat Status Pengajuan Mobil (1).



Gambar 92. Hasil *Automated Testing* Untuk *Use Case* Melihat Status Pengajuan Mobil (2).

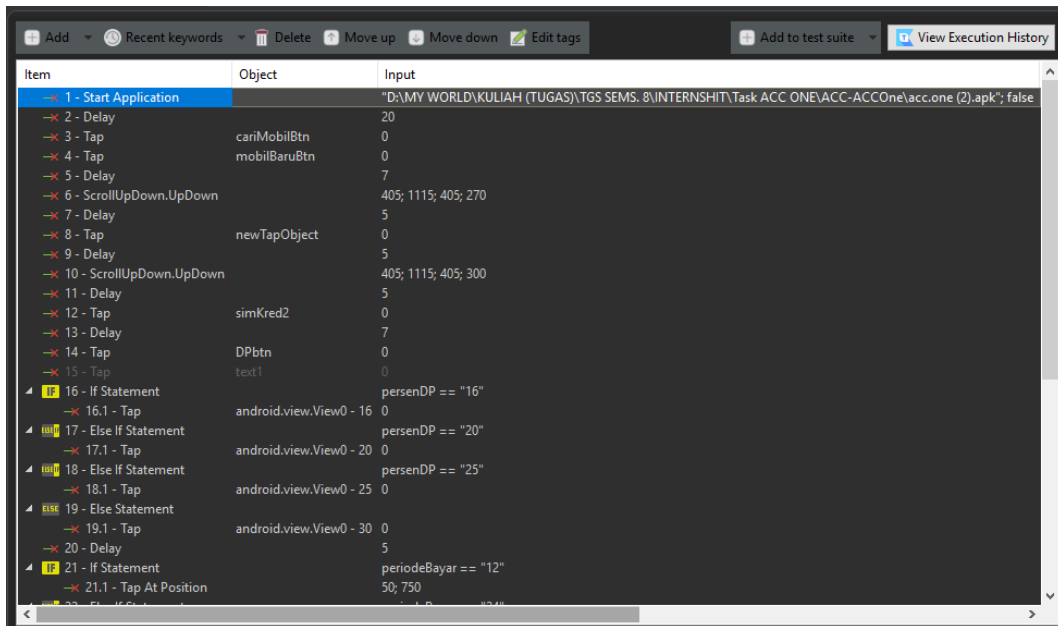


```
03-09-2020 12:40:19 PM Test Suites/Lihat Status Pengajuan
Elapsed time: 6m - 28,334s
```

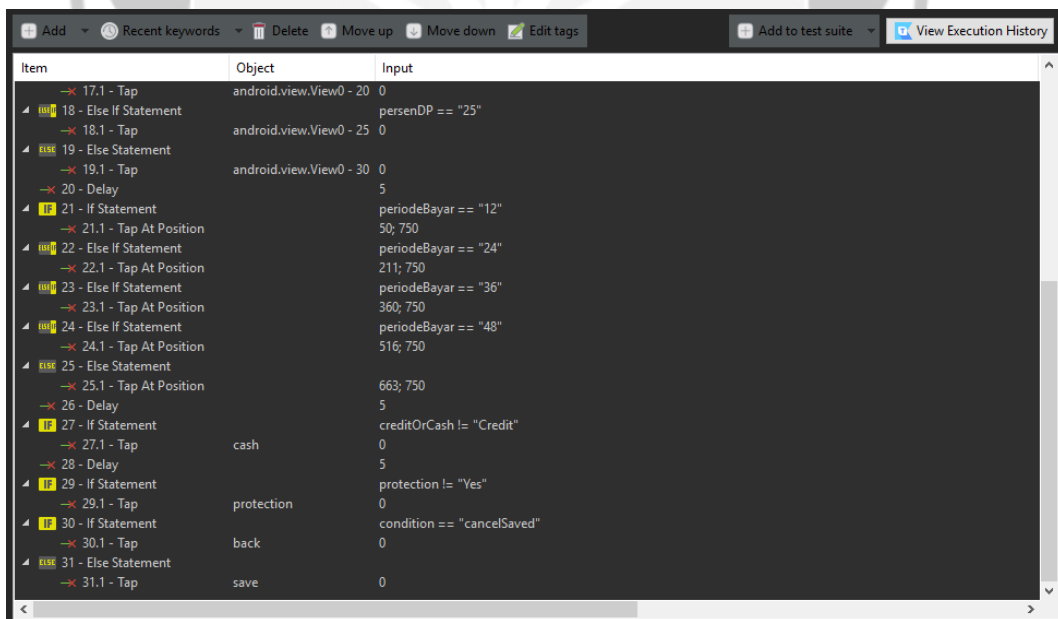
Gambar 93. Hasil *Automated Testing* Untuk *Use Case* Melihat Status Pengajuan Mobil (3).

Pada Gambar 89, 90, dan 91, ditunjukkan hasil eksekusi pengujian otomatis *test case* melihat status pengajuan mobil. Waktu yang dibutuhkan untuk mengeksekusi *test suite* ini dengan 4 kali pengujian dan 4 *data binding* adalah 6 menit 28 detik. Hasil dari keseluruhan pengujian ini adalah *passed* dengan persentase kesesuaian fungsional 100% di mana dapat dikatakan tidak ditemukan *bug* pada fungsi yang diuji.

5.2.2.9 Penerapan *Automated Testing* Pada Katalon Untuk *Use Case* Simulasi Kredit Mobil Baru (Paket)



Gambar 94. *Test Case* Katalon Untuk *Use Case* Simulasi Kredit Mobil Baru (Paket) (1).



Gambar 95. *Test Case* Katalon Untuk *Use Case* Simulasi Kredit Mobil Baru (Paket) (2).

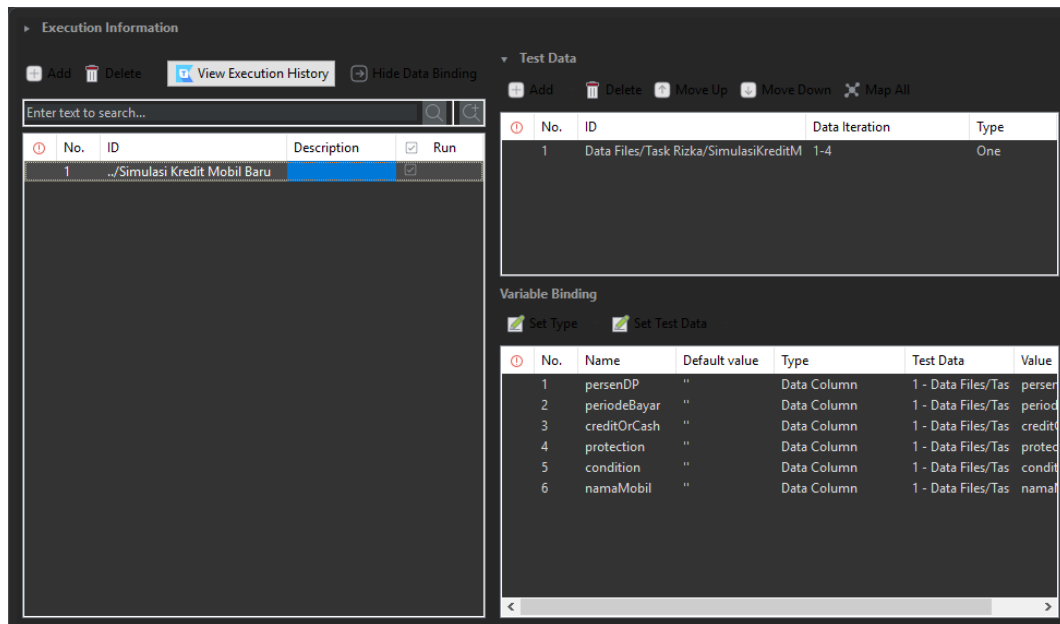
Pada Gambar 92 dan 93, merupakan rancangan *test case* untuk menguji fungsi simulasi kredit mobil baru (paket). Fungsi ini diuji pada aplikasi acc.one

dengan versi *update* per tanggal 12 Februari 2020. Pada dasarnya, fungsi simulasi kredit mobil baru (paket) ini tergolong ke dalam *use case* mengelola pembeli mobil *online* yang dijabarkan pada Gambar 3. Alur pengujian yang diterapkan untuk *automated testing* ini sesuai dengan alur pengujian manual. Pada *test case* terdapat instruksi *Tap At Position* yang berfungsi untuk *tap slider* periode bayar dikarenakan objek *slider* tidak dapat berfungsi jika diuji menggunakan Katalon, sebagai gantinya penggunaan instruksi tersebut dapat bekerja karena *tap* dilakukan dengan menentukan posisi x dan y seperti yang ditunjukkan pada Gambar 93 poin 21.1, 22.1, 23.1, 24.1, dan 25.1. Pembuatan objek baru dilakukan untuk dapat *tap* berdasarkan teks variabel *namaMobil*. Pengaturan pembuatan objek baru sama dengan Gambar 19 dan 20 dengan variabel yang disesuaikan dengan kebutuhan.

No.	Name	Type	Default value
1	namaMobil	String	''
2	persenDP	String	''
3	periodeBayar	String	''
4	creditOrCash	String	''
5	protection	String	''
6	condition	String	''

Gambar 96. Variabel Katalon Untuk *Use Case* Simulasi Kredit Mobil Baru (Paket).

Pada Gambar 94, ditunjukkan variabel-variabel yang digunakan pada pengujian otomatis di Katalon untuk *use case* simulasi kredit mobil baru (paket). Variabel *condition* digunakan sebagai alat untuk membantu menentukan alur alternatif, sehingga alur pengujian dapat lebih singkat. Variabel *condition* juga dapat digunakan untuk mengurangi penggunaan *if-else* di dalam *if-else*. Dengan membuat kondisi yang lebih bervariasi, pembuatan *test case* nantinya akan lebih singkat dan mudah dipahami dengan mengandalkan nilai dari variabel *condition* sebagai kondisi di *if-else* dalam penentuan alur pengujian.



Gambar 97. Test Suite Katalon Untuk Use Case Simulasi Kredit Mobil Baru (Paket).

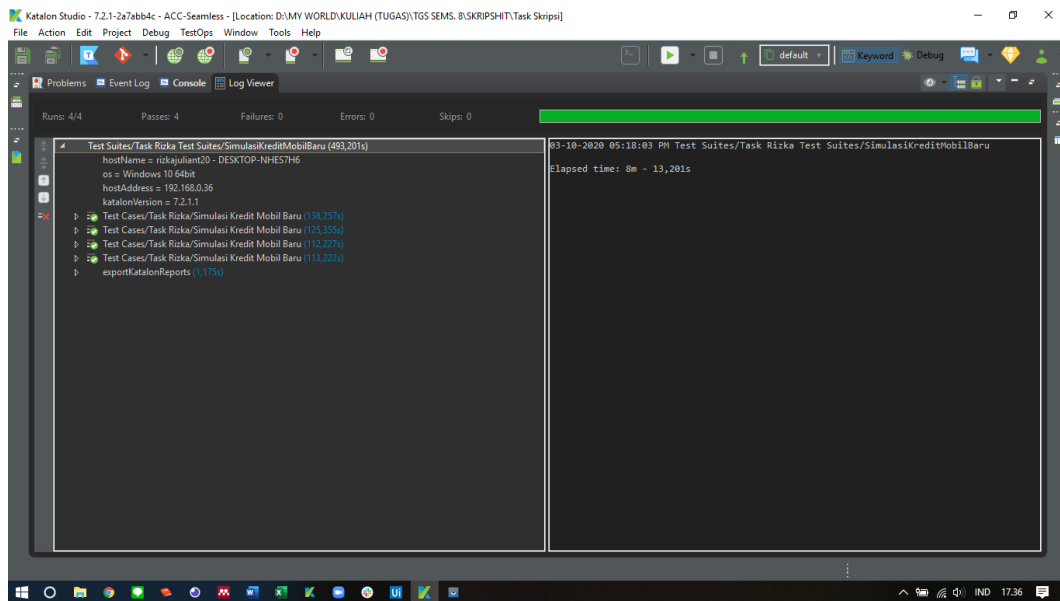
Test suite yang ditunjukkan pada Gambar 95, berfungsi untuk menjalankan test case simulasi kredit mobil baru (paket) yang telah dibuat sebelumnya. Pada test suite, variabel-variabel yang sudah diinisialisasi, dipasangkan dengan data binding pada Microsoft Excel, agar nilai yang dimasukkan pada saat pengujian sesuai.

	A	B	C	D	E	F	G
1	namaMobil	persenDP	periodeBaya	creditOrCash	protection	condition	
2	TOYOTA AGYA 2019 BENSIN	16	24	Credit	No	passed	
3	TOYOTA AGYA 2019 BENSIN	16	24	Credit	Yes	passed	
4	TOYOTA AGYA 2019 BENSIN	25	48	Cash	Yes	cancelSaved	
5	TOYOTA AGYA 2019 BENSIN	25	48	Cash	No	cancelSaved	

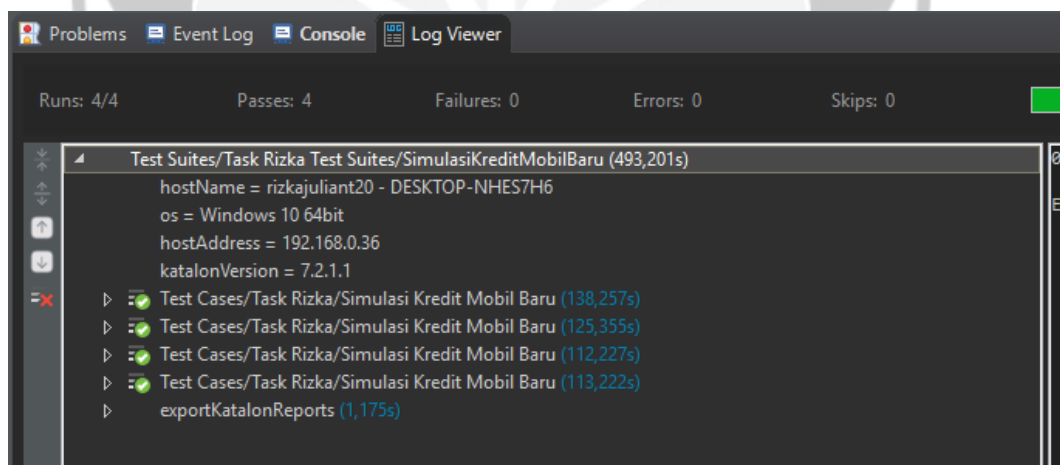
Gambar 98. Data Binding Untuk Use Case Simulasi Kredit Mobil Baru (Paket).

Pada Gambar 96, ditunjukkan data binding yang dibuat pada Microsoft Excel, dengan nilai aktual yang disesuaikan dengan pengujian manual, di mana file ini akan dimasukkan ke dalam Katalon. Terdapat 4 data binding yang berarti akan dilakukan 4 kali pengujian dengan variasi skenario testing yang berbeda-beda untuk mendeteksi apakah terjadi bug dengan variasi skenario pengujian bernilai positif dan negatif. Pada data binding ini, terdapat kondisi passed, di mana alur yang dijalankan akan mencapai post condition, sedangkan kondisi

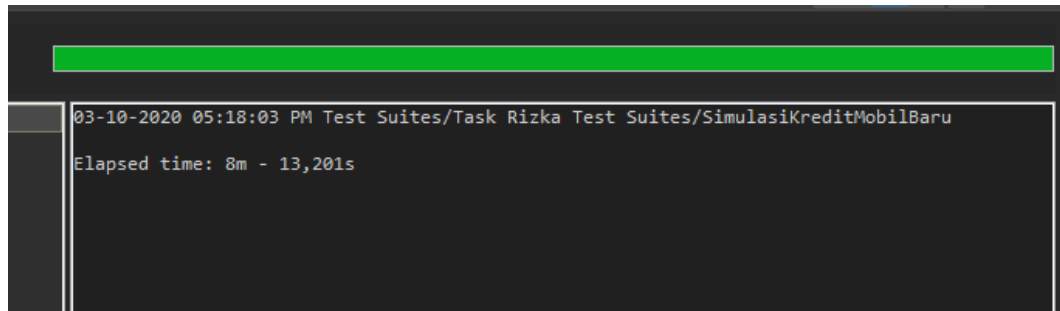
cancelSaved merupakan kondisi yang akan menjalankan sebaliknya yaitu pembatalan penyimpanan simulasi kredit.



Gambar 99. Hasil *Automated Testing* Untuk *Use Case* Simulasi Kredit Mobil Baru (Paket) (1).



Gambar 100. Hasil *Automated Testing* Untuk *Use Case* Simulasi Kredit Mobil Baru (Paket) (2).

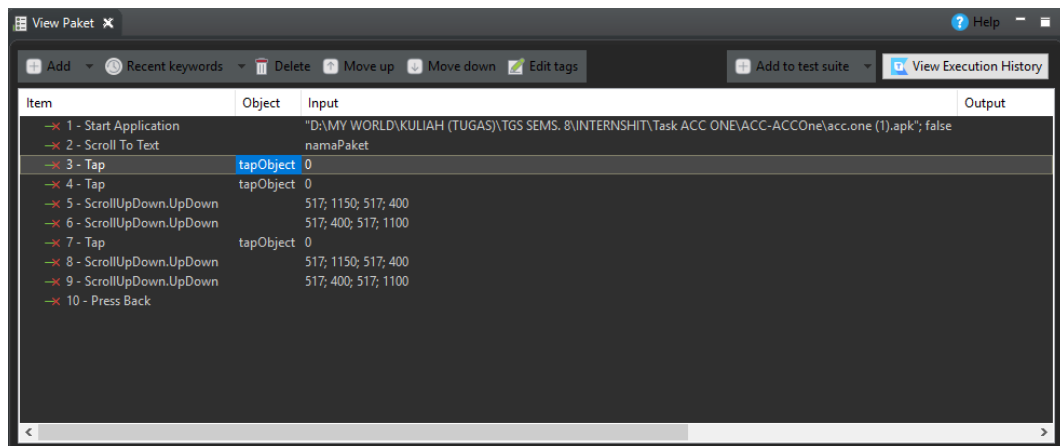
A screenshot of a terminal window showing the results of an automated test. The text in the terminal is: "03-10-2020 05:18:03 PM Test Suites/Task Rizka Test Suites/SimulasiKreditMobilBaru" followed by "Elapsed time: 8m - 13,201s". The terminal has a black background with white text and a green progress bar at the top.

```
03-10-2020 05:18:03 PM Test Suites/Task Rizka Test Suites/SimulasiKreditMobilBaru
Elapsed time: 8m - 13,201s
```

Gambar 101. Hasil *Automated Testing* Untuk *Use Case* Simulasi Kredit Mobil Baru (Paket) (3).

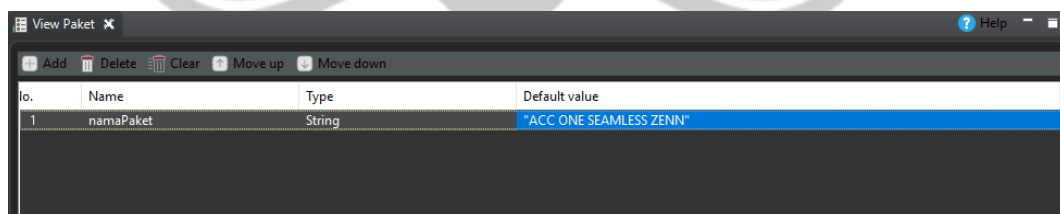
Pada Gambar 97, 98, dan 99, ditunjukkan hasil eksekusi pengujian otomatis *test case* simulasi kredit mobil baru (paket). Waktu yang dibutuhkan untuk mengeksekusi *test suite* ini dengan 4 kali pengujian dan 4 *data binding* adalah 8 menit 13 detik. Hasil dari keseluruhan pengujian ini adalah *passed* dengan persentase kesesuaian fungsional 100% di mana dapat dikatakan tidak ditemukan *bug* pada fungsi yang diuji.

5.2.2.10 Penerapan *Automated Testing* Pada Katalon Untuk *Use Case* Melihat Paket Pembelian Mobil



Gambar 102. *Test Case* Katalon Untuk *Use Case* Melihat Paket Pembelian Mobil.

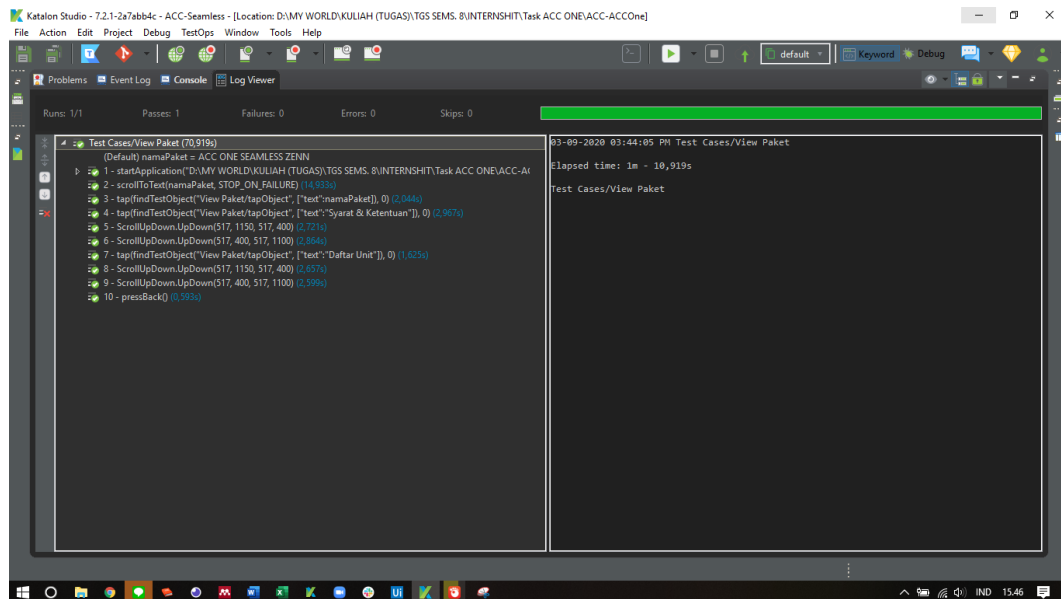
Pada Gambar 100, merupakan rancangan *test case* untuk menguji fungsi melihat paket pembelian mobil. Fungsi ini diuji pada aplikasi acc.one dengan versi *update* per tanggal 12 Februari 2020. Pada dasarnya, fungsi mencari promo mobil tertentu ini tergolong ke dalam *use case* mengelola pembelian mobil *online* yang dijabarkan pada Gambar 3. Alur pengujian yang diterapkan untuk *automated testing* ini sesuai dengan alur pengujian manual. Pada *test case* ini dibuat objek baru guna mendeteksi objek berdasarkan teks dari variabel namaMobil. *Test case* ini menggunakan instruksi *custom keyword scrollUpDown* untuk melihat detail dari paket mobil.



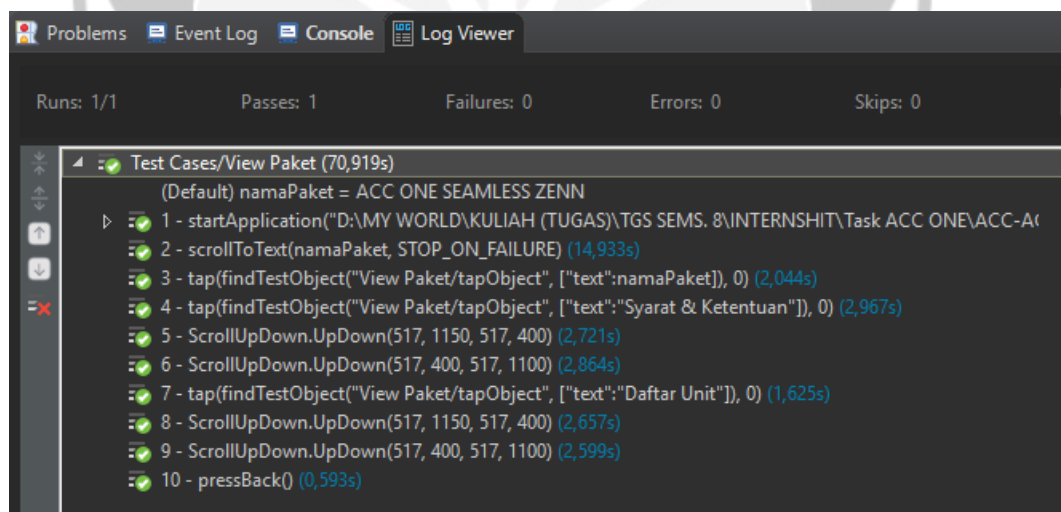
Gambar 103. Variabel Katalon Untuk *Use Case* Melihat Paket Pembelian Mobil.

Pada Gambar 101, ditunjukkan variabel-variabel yang digunakan pada pengujian otomatis di Katalon untuk *use case* mencari promo mobil tertentu. Berbeda dari *test case* sebelumnya, untuk *use case* ini tidak menggunakan *data binding* yang dibuat di Microsoft Excel dan juga tidak menggunakan *test suite*

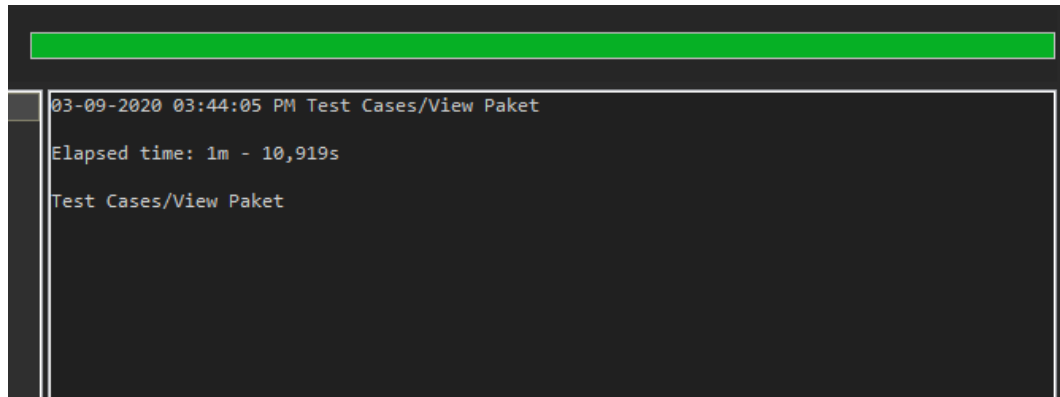
karena tidak terdapat alur alternatif pada perancangannya. Terdapat 1 variabel yang digunakan yaitu namaPaket yang nilainya diisi langsung pada *test case*.



Gambar 104. Hasil *Automated Testing* Untuk *Use Case* Melihat Paket Pembelian Mobil (1).



Gambar 105. Hasil *Automated Testing* Untuk *Use Case* Melihat Paket Pembelian Mobil (2).



```
03-09-2020 03:44:05 PM Test Cases/View Paket
Elapsed time: 1m - 10,919s
Test Cases/View Paket
```

Gambar 106. Hasil *Automated Testing* Untuk *Use Case* Melihat Paket Pembelian Mobil (3).

Pada Gambar 102, 103, dan 104, ditunjukkan hasil eksekusi pengujian otomatis *test case* melihat paket pembelian mobil. Waktu yang dibutuhkan untuk mengeksekusi *test case* ini dengan 1 kali pengujian adalah 1 menit 10 detik. Hasil dari keseluruhan pengujian ini adalah *passed* dengan persentase kesesuaian fungsional 100% di mana dapat dikatakan tidak ditemukan *bug* pada fungsi yang diuji.

5.3 Pembahasan Hasil Perbandingan Pengujian Manual dan Otomatis

Berdasarkan pengujian yang telah dilakukan pada sub bab sebelumnya, ditemukan *bug* atau kelainan di 2 fungsi utama yang diuji pada aplikasi *mobile acc.one* yang membuktikan bahwa aplikasi masih harus diperbaiki demi terjaganya kualitas di tangan *customer ACC*. Daripada itu, terdapat pula beberapa kendala yang dihadapi pada saat berjalannya pengujian manual dan otomatis aplikasi *mobile acc.one*, antara lain:

- a. Seringnya *update* aplikasi yang dilakukan oleh pengembang dan banyaknya perubahan objek pada aplikasi membuat *test case* perlu untuk juga di *update*. *Update* aplikasi yang dilakukan pengembang dikarenakan peluncurannya yang baru, dan aplikasi *acc.one* merupakan salah satu aplikasi terbesar yang digunakan oleh ACC untuk menarik *customer*.
- b. Masih banyak objek-objek yang tidak terdeteksi ketika pengujian otomatis menggunakan Katalon, sehingga harus mencari alternatif lain untuk menggantikan peran objek-objek yang tidak dapat terdeteksi tersebut salah satunya dengan membuat objek baru.
- c. Terbatasnya instruksi di Katalon yang dapat digunakan sehingga diperlukan pembuatan *custom keyword* dengan cara merancang kode secara manual dengan menggunakan bahasa Groovy, seperti tertera pada Gambar 105 yang merupakan *keyword* baru untuk *scroll* ke atas dan ke bawah.

```
import com.kms.katalon.core.annotation.Keyword

public class ScrollUpDown {

    @Keyword
    def UpDown(int startX, int startY, int endX, int endY){

        def device_Height = Mobile.getDeviceHeight()
        def device_Width = Mobile.getDeviceWidth()

        startX = device_Width / 2
        endX = startX

        startY = device_Height * 0.30
        endY = device_Height * 0.70

        Mobile.swipe(startX, endY, endX, startY)
    }
}
```

Gambar 107. Custom Keyword Untuk Scroll Up Down.

- d. Sering terjadi kesalahan pada saat *input* data dan salah melakukan alur skenario pada saat pengujian secara manual, sehingga pengujian harus diulangi sesuai dengan skenario dan nilai aktual yang telah disusun pada Bab 4 dan 5.
- e. Memori perangkat uji, harus dapat memuat aplikasi yang ingin diuji dan juga aplikasi Appium Settings. Jika memori tidak dapat memenuhi kapasitas aplikasi tersebut, maka Katalon tidak dapat menguji aplikasi.

Tabel 12. Perbandingan Waktu Antara Pengujian Manual dan Pengujian Otomatis.

No.	Nama Fungsi	Waktu Pengujian			
		Manual	Otomatis (Katalon)	Selisih Waktu	Persentase Kecepatan Pengujian Manual Dibandingkan Otomatis
1	Mencari Mobil Baru	5 menit 1 detik	10 menit 25 detik	5 menit 24 detik	$(324/625)*100 = 51,84\%$
2	Mencari Promo Mobil Tertentu	1 menit 4 detik	2 menit 33 detik	1 menit 29 detik	$(89/153)*100 = 58,16\%$
3	Mencari Mobil Bekas (Marketplace)	5 menit 17 detik	10 menit 35 detik	5 menit 18 detik	$(318/635)*100 = 50,07\%$
4	Mencari Mobil Bekas (Lelang)	4 menit 48 detik	7 menit 19 detik	2 menit 31 detik	$(151/439)*100 = 34,39\%$
5	Mengubah Foto Profil	1 menit 44 detik	3 menit 50 detik	2 menit 6 detik	$(126/230)*100 = 54,78\%$
6	Mencari Mobil Rental	5 menit 52 detik	12 menit 13 detik	6 menit 21 detik	$(381/733)*100 = 51,98\%$
7	Lupa Kata Sandi	10 menit 8 detik	22 menit 36 detik	12 menit 28 detik	$(748/1356)*100 = 55,16\%$

8	Melihat Status Pengajuan Mobil	1 menit 57 detik	6 menit 28 detik	4 menit 31 detik	$(271/388)*100 = 69,84\%$
9	Simulasi Kredit Mobil Baru (Kredit)	3 menit 17 detik	8 menit 13 detik	4 menit 56 detik	$(296/493)*100 = 60,04\%$
10	Melihat Paket Pembelian Mobil	24 detik	1 menit 10 detik	46 detik	$(46/70)*100 = 65,71\%$
Rata - Rata Persentase Kecepatan Pengujian Manual Dibandingkan Pengujian Otomatis					55,19%

Dalam tujuan disebutkan bahwa terdapat pula perbandingan waktu yang dicatat pada eksperimen pengujian ini. Maka dari itu, telah dicatat waktu dibutuhkan saat pengujian manual dan otomatis untuk masing-masing fungsi yang dijabarkan pada Tabel 12, dan dapat disimpulkan bahwa pengujian manual lebih cepat dibandingkan dengan pengujian otomatis.

Hal ini dibuktikan berdasarkan selisih waktu dan persentase kecepatan pengujian manual dibandingkan dengan otomatis setelah dilakukan perhitungan. Perhitungan persentase kecepatan pengujian, dihitung menggunakan rumus:

$$\text{Persentase Kecepatan} = \frac{|(\text{Waktu Manual}) - (\text{Waktu Otomatis})|}{(\text{Waktu Otomatis})} \times 100\%$$

Dan terdapat pula perhitungan rata-rata persentase kecepatan pengujian manual dibandingkan otomatis yang dihitung menggunakan rumus:

$$\text{Rata - Rata Kecepatan} = \frac{(\text{Jumlah Persentase Dari Masing - Masing Fungsi})}{(\text{Jumlah Fungsi})}$$

Di mana menghasilkan rata-rata sebesar 55,19% lebih cepat jika dilakukan dengan pengujian secara manual. Hasil tersebut dapat terjadi karena adanya faktor yang mempengaruhi lamanya pengujian otomatis. Salah satu faktornya

yaitu Katalon yang membutuhkan waktu untuk mendeteksi objek, sehingga terdapat *delay*. Seperti pada instruksi *Verify Element Visible* yang di mana Katalon akan mendeteksi objek terlebih dahulu, selanjutnya baru akan memverifikasi bahwa objek tersedia yang kedua hal tersebut cukup memakan waktu. Jika objek tidak dapat terdeteksi, maka jalannya pengujian akan terganggu. Hal yang dapat menyebabkan objek tidak dapat terdeteksi lainnya adalah pada saat terjadinya gangguan koneksi internet yang mengakibatkan data tidak dapat dimuat oleh aplikasi, sehingga Katalon tidak dapat mendeteksi objek apa pun dan pengujian akan gagal. Maka dari itu, Katalon menyediakan instruksi *delay* itu sendiri yang dapat diatur lamanya, sehingga penguji yang sudah pernah menganalisis alur pengujian secara manual, dapat memperkirakan lama waktu tunggu aplikasi untuk memuat data dan mengimplementasikan instruksi tersebut pada Katalon. Hal tersebut berbanding terbalik ketika menggunakan cara manual, karena lama waktu pemuatan data di aplikasi tidak akan mempengaruhi jalannya pengujian, sehingga penguji dapat menunggu data dimuat terlebih dahulu, lalu barulah dapat lanjut ke langkah pengujian berikutnya tanpa harus gagal.

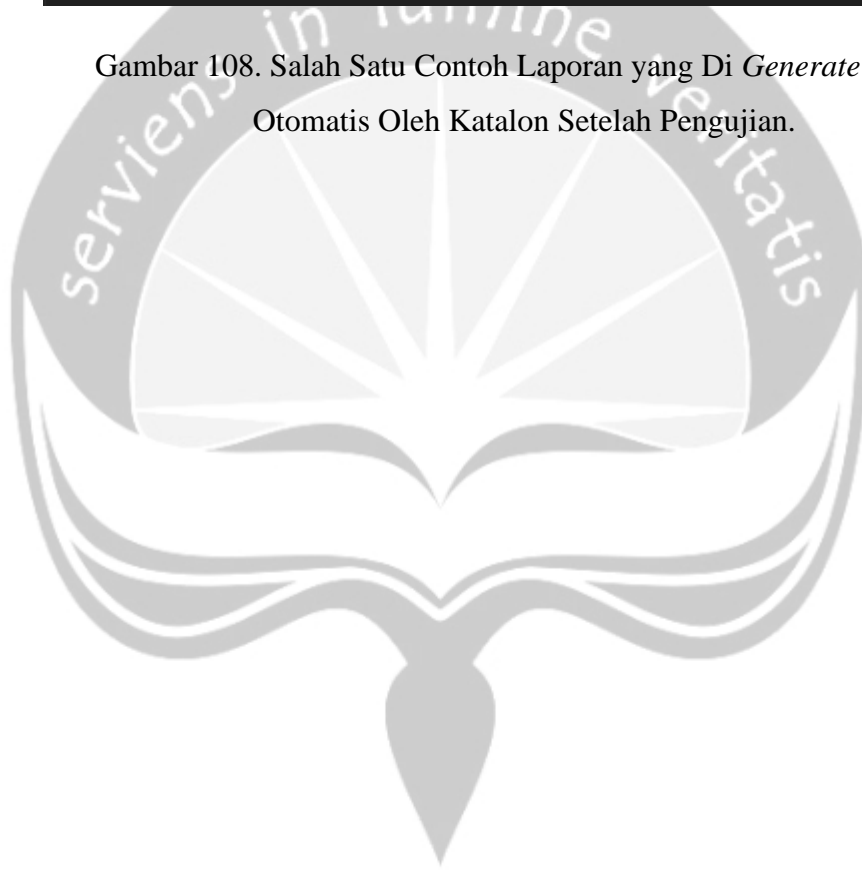
Di luar dari hasil perbandingan pengujian secara manual dan otomatis berdasarkan waktu, terdapat pula keuntungan pengujian otomatis menggunakan Katalon, antara lain:

- a. *Test case* dapat *diupdate* dan digunakan kembali, sehingga dapat mengikuti perkembangan dari aplikasi yang diuji itu sendiri dan tidak memakan waktu yang banyak seperti pembuatan *test case* dari awal.
- b. Meringankan pekerjaan penguji apabila *data binding* yang disediakan untuk pengujian terlampau banyak, sehingga penguji dapat fokus pada perancangan alur *test case* dan hasil pengujian untuk nantinya dilaporkan kepada tim pengembang apabila ditemukan *bug* atau *error*.

- c. Hasil pengujian sudah dapat di *generate* secara otomatis ke dalam bentuk *soft file* dengan format-format seperti Gambar 106 di bawah ini.

Name	Date modified	Type	Size
20200318_131101	18/03/2020 13.17	Microsoft Excel C...	4 KB
20200318_131101	18/03/2020 13.17	AVG HTML Docu...	408 KB
1584512268493	18/03/2020 13.17	PNG File	72 KB
1584512270024	18/03/2020 13.17	PNG File	72 KB
appium	18/03/2020 13.17	Text Document	196 KB
execution.properties	18/03/2020 13.11	PROPERTIES File	2 KB
execution.uuid	18/03/2020 13.17	UUID File	1 KB
execution0	18/03/2020 13.17	Text Document	144 KB
execution0.log.lck	18/03/2020 13.11	LCK File	0 KB
JUnit_Report	18/03/2020 13.17	XML Document	47 KB
testCaseBinding	18/03/2020 13.11	File	2 KB

Gambar 108. Salah Satu Contoh Laporan yang Di *Generate* Secara Otomatis Oleh Katalon Setelah Pengujian.



BAB VI. PENUTUP

6.1 Kesimpulan

Terdapat beberapa kesimpulan yang dapat diambil berdasarkan penelitian kali ini, antara lain:

1. Berdasarkan perhitungan waktu yang dibutuhkan untuk pengujian, cara manual lebih cepat daripada pengujian otomatis menggunakan Katalon. Namun, terdapat beberapa kelemahan yang didapatkan dari penggunaan cara manual ini seperti banyaknya kesalahan pada saat *input* data uji dan banyaknya keterlibatan penguji pada saat proses pengujian dari pembuatan *test case*, eksekusi, hingga pelaporan yang serba manual.
2. Pengujian otomatis pada dasarnya diciptakan untuk meringankan beban dari pada penguji. Proses pelaporan yang terstruktur dan otomatis berdampak pada minimalnya keterlibatan penguji di dalam proses pengujian. Proses memasukkan data uji secara otomatis melalui Katalon juga meminimalkan kesalahan yang biasa dilakukan pada saat pengujian manual. *Test case* yang telah dibuat juga dapat digunakan kembali, dan dapat diubah sewaktu-waktu jika terjadi *update* pada aplikasi yang diuji. Kekurangan dari penggunaan *tool* Katalon ini terdapat pada lamanya waktu yang dibutuhkan untuk pendeteksian objek dan juga terbatasnya instruksi pada Katalon yang memungkinkan penguji untuk membuat *custom keyword* sendiri dengan merancang *code* menggunakan bahasa pemrograman Groovy.
3. Berdasarkan pengujian secara manual dan otomatis terhadap 10 fungsi utama aplikasi *mobile acc.one*, ditemukan 2 *bug* atau kesalahan pada 2 fungsi, antara lain fungsi Mencari Mobil Bekas (*Marketplace*) dan Lupa Kata Sandi.

6.2 Saran

Untuk penelitian selanjutnya disarankan untuk menambahkan parameter banyaknya data uji sebagai acuan perbandingan keefektifan pengujian otomatis dan manual. Diharapkan dengan adanya parameter banyaknya data uji, dapat menunjukkan pengaruhnya terhadap waktu yang dibutuhkan untuk

pengujian manual dan otomatis. Parameter waktu dan banyaknya data uji dapat diproyeksikan dengan grafik, dengan harapan semakin banyak data uji, waktu yang dibutuhkan untuk pengujian manual dan otomatis akan bertambah pula, namun fokusnya ada pada persentase penurunan waktu yang dibutuhkan yang diharapkan pengujian otomatis akan lebih sedikit persentase penurunannya dibandingkan dengan manual, sehingga pengujian otomatis dapat dikatakan lebih efektif.



DAFTAR PUSTAKA

- [1] “Pengguna Smartphone di Indonesia 2016-2019,” 2020. [Online]. Available:
<https://databoks.katadata.co.id/datapublish/2016/08/08/pengguna-smartphone-di-indonesia-2016-2019>. [Accessed: 03-Feb-2020].
- [2] A. Kaur, “Review of Mobile Applications Testing with Automated Techniques,” *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 4, no. 10, pp. 503–507, 2015.
- [3] G. Lovreto, A. T. Endo, P. Nardi, and V. H. S. Durelli, “Automated Tests for Mobile Games: An Experience Report,” *Brazilian Symp. Games Digit. Entertain. SBGAMES*, vol. 2018-Novem, no. October, pp. 48–56, 2019.
- [4] H. Anjum *et al.*, “A Comparative Analysis of Quality Assurance of Mobile Applications using Automated Testing Tools,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 7, 2017.
- [5] A. Méndez-Porras, C. Quesada-López, and M. Jenkins, “Automated Testing of Mobile Applications: A Systematic Map and Review,” *CIBSE 2015 - XVIII Ibero-American Conf. Softw. Eng.*, no. April, pp. 195–208, 2015.
- [6] T. Mikkonen, “RISTO AUTIO AUTOMATED TESTING OF CROSS-PLATFORM MOBILE APPLICATIONS,” no. February, 2016.
- [7] U. Hanifah, R. Alit, and S. Sugiarto, “Penggunaan Metode Black Box Pada Pengujian Sistem Informasi Surat Keluar Masuk,” *SCAN - J. Teknol. Inf. dan Komun.*, vol. 11, no. 2, pp. 33–40, 2016.
- [8] M. Mursyidah and H. T. Hidayat, “Pengujian Sistem Informasi Akuntansi Biaya Operasional Sekolah Dengan Black Box Testing,” *J. Infomedia*, vol. 2, no. 2, pp. 7–14, 2018.
- [9] S. Janti, “Animasi Edukasi Interaktif Tes Kemampuan Konsentrasi Dengan Permainan Tebak Warna,” *J. Tek. Komput. AMIK BSI*, vol. III, no. 1, pp.

106–113, 2017.

- [10] H. B. I. Alfaris, C. Anam, and A. Masy'an, "IMPLEMENTASI BLACK BOX TESTING PADA SISTEM INFORMASI PENDAFTARAN SANTRI BERBASIS WEB DENGAN MENGGUNAKAN PHP DAN MYSQL," *SAINTEKBU J. Sains dan Teknol.*, vol. 6, no. 1, pp. 23–38, 2013.
- [11] W. N. Cholifah, Y. Yulianingsih, and S. M. Sagita, "Penguujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap," *STRING (Satuan Tulisan Ris. dan Inov. Teknol.*, vol. 3, no. 2, p. 206, 2018.
- [12] M. S. Mustaqbal, R. F. Firdaus, and H. Rahmadi, "PENGUJIAN APLIKASI MENGGUNAKAN BLACK BOX TESTING BOUNDARY VALUE ANALYSIS (Studi Kasus: Aplikasi Prediksi Kelulusan SNMPTN)," vol. I, no. 3, pp. 31–36, 2015.
- [13] T. S. Jaya, "Penguujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung)," *J. Inform. Pengemb. IT*, vol. 3, no. 2, pp. 45–48, 2018.
- [14] M. K. MZ, "PENGUJIAN PERANGKAT LUNAK METODE BLACK-BOX BERBASIS EQUIVALENCE PARTITIONS PADA APLIKASI SISTEM INFORMASI SEKOLAH," *J. Mikrotik*, vol. 06, no. 03, pp. 02–16, 2016.
- [15] Y. Irawan, S. Muzid, N. Susanti, and R. Setiawan, "System Testing Using Black Box Testing Equivalence Partitioning (Case Study at Garbage Bank Management Information System on Karya Sentosa)," pp. 1–7, 2019.
- [16] F. C. Ningrum, D. Suherman, S. Aryanti, and H. A. Prasetya, "Penguujian Black Box pada Aplikasi Sistem Seleksi Sales Terbaik Menggunakan Teknik Equivalence Partitions," vol. 4, no. 4, pp. 125–130, 2020.
- [17] A. C. Barus and L. Siburian, "STUDI PERBANDINGAN ALAT PENGUJIAN OTOMATIS UNTUK APLIKASI ANDROID," *Teknol. Inf.*

dan Ilmu Komput., vol. 6, no. 6, pp. 645–654, 2019.

- [18] Herlinda, D. Katarina, and E. W. Ambarsari, “AUTOMATION TESTING TOOL DALAM PENGUJIAN APLIKASI BELAJAR TAJWID PADA PLATFORM ANDROID,” *STRING (Satuan Tulisan Ris. dan Inov. Teknol.*, vol. 4, no. 2, pp. 205–212, 2019.
- [19] M. L. Larrea, “Black-Box Testing Technique for Information Visualization . Sequencing Black-Box Testing Technique for Information Visualization . Sequencing Constraints with Low-Level Interactions,” *Comput. Sci. Technol.*, vol. 17, no. April, pp. 37–48, 2017.
- [20] N. Gupta, “Different Approaches to White Box Testing to Find Bug,” *Int. J. Adv. Res. Comput. Sci. Technol. (IJARCST 2014)*, vol. 2, no. 3, pp. 46–49, 2014.
- [21] “Gambar Fitur Aplikasi ACC ONE.” [Online]. Available: <https://www.acc.co.id/images/2019/img-accone.jpg>. [Accessed: 03-Feb-2020].
- [22] “ACC Luncurkan acc.one, Aplikasi Mobile Terlengkap Untuk Pembiayaan,” 2019. [Online]. Available: <https://www.acc.co.id/corporatenews/read/acc-luncurkan-acc-one-aplikasi-mobile-terlengkap-untuk-pembiayaan>. [Accessed: 19-Jan-2020].
- [23] “Welcome to Katalon Studio,” 2020. [Online]. Available: <https://docs.katalon.com/katalon-studio/docs/overview.html>. [Accessed: 19-Jan-2020].
- [24] Katalon, “Android setup on Windows,” 2020. [Online]. Available: https://docs.katalon.com/katalon-studio/videos/android_setup_windows.html. [Accessed: 20-Apr-2020].
- [25] “Introduction to Appium.” [Online]. Available: <http://appium.io/docs/en/about-appium/intro/>. [Accessed: 27-Mar-2020].