JEPIN
(Jurnal Edukasi dan Penelitian Informatika)
ISSN(e): 2548-9364 / ISSN(p) : 2460-0741

Vol. 7
No. 3
Desember
2021

Pemanfaatan *Katalon Studio* untuk Otomatisasi Pengujian *Black-Box* pada Aplikasi iPosyandu

Arief Zulianto[#1], Ayi Purbasari[*2], Neni Suryani[*3], Ari Indra Susanti[+4], Fedri R. Rinawan[+5], Wanda G. Purnama[*6]

[#]*Magister Teknik Informatika, Universitas Langlangbuana*
Jalan Karapitan No. 116, Kota Bandung, Jawa Barat
[1]madzul@unla.ac.id

[*]*Teknik Informatika, Universitas Pasundan*
Jl. Dr. Setiabudi No.193, Gegerkalong, Kec. Sukasari, Kota Bandung, Jawa Barat 40153
[2]pbasari@unpas.ac.id
[3]neni.suryani@gmail.com
[6]wanda.gusdya@unpas.ac.id

[+]*Dept. Ilmu Kesehatan Masyarakat, Pusat Studi Sistem Kesehatan dan Inovasi Pendidikan Tenaga Kesehatan Universitas Padjadjaran Jalan Professor Eyckman No.3 Kota Bandung, Jawa Barat,*
[4]ari.indra@unpad.ac.id
[5]f.rinawan@unpad.ac.id

Abstrak— **iPosyandu merupakan aplikasi berbasis mobile dan web yang dirancang untuk memudahkan melihat aktivitas di posyandu dan menyusun laporan aktivitas tersebut. Aplikasi ini dibangun sejak 2018 dan terus dikembangkan seiring dengan luasnya pengguna IPosyandu yang ditargetkan di seluruh Indonesia. Oleh karena itu diperlukan peningkatan kualitas dari aplikasi iPosyandu melalui proses pengujian. Penelitian ini menerapkan kakas Katalon Studio untuk mengotomatisasi proses pengujian *black box* aplikasi iPosyandu. Pengujian bertujuan untuk meminimalisir apa yang tidak bisa di *back up* oleh pengujian secara manual dan menghindari *human error*. *Test case* yang digunakan dalam pengujian dengan Katalon Studio menerapkan *record* dan *playback*. Pengujian otomatis ini dibandingkan dengan manual dan menunjukkan hasil yang sama, dengan waktu pengujian yang lebih cepat. Untuk 13 kasus uji yang dicobakan, diperoleh peningkatan kecepatan hasil eksekusi menjad 283,08 detik dibandingkan dengan 719,27detik ketika diuji secara manual. Dengan demikian terjadi peningkatan kecepatan 2,54 kali.**

Kata kunci*— *iPosyandu, Pengujian, Automation Testing Tool, Katalon Studio, Black-Box, Test Case

I. PENDAHULUAN

iPosyandu adalah aplikasi berbasis Android dan berbasis Web sejak tahun 2018, yang dibuat untuk pencatatan dan pelaporan Sistem Informasi Posyandu (Pos Pelayanan Kesehatan Terpadu) bulanan dan tahunan yang semula berupa buku besar tertulis manual [1]. Aplikasi dibangun dengan pendekatan pola kerja kader [2] yang menjadi pengguna aplikasi ini [3]. iPosyandu merupakan aplikasi yang dirancang agar lebih memudahkan kader untuk menyusun laporan yang berupa aktivitas di Posyandu kemudian menyusun laporannya. Aplikasi iPosyandu pun bisa digunakan sebagai memantau aktivitas tumbuh kembang anak yang dapat berkoneksi khusus antara kader dengan orang tua [3] . Aplikasi ini disertai dengan manual dan monitoring feedback dari kader yang menjadi pengguna [4]. Aplikasi ini saat ini digunakan oleh kader di Posyandu Kecamatan Pasawahan Kabupaten Purwakarta dan akan diperluas ke Posyandu binaan PT Astra International, Tbk yang tersebar di seluruh Indonesia. Dengan demikian, aplikasi iPosyandu memiliki data yang besar dengan pengguna akhir kader di berbagai daerah di Indonesia. Seiring dengan perluasaan penggunaan iPosyandu, semakin diperlukan jaminan bahwa aplikasi ini memiliki fungsionalitas yang sesuai dengan kebutuhan pengguna. Untuk itu, pengujian menjadi hal yang harus dilakukan secara intensif dan menyeluruh.

Membangun atau memelihara sebuah aplikasi tidak terlepas dari dilakukannya pengujian. Banyak organisasi perangkat lunak menghabiskan hingga 40% dari sumber daya mereka untuk pengujian [5]. Pengujian lebih dari sekadar *debugging*. Pengujian tidak hanya digunakan untuk menemukan dan memperbaiki cacat. Tetapi juga digunakan dalam validasi, proses verifikasi dan pengukuran reliabilitas [6]. Terdapat berbagai pendekatan dalam pengujian, antara lain berdasarkan teknik pengujian *black-box* dan *white-box* atau melihat tingkatan pengujian itu sendiri, yaitu pengujian unit, integrasi ataupun pengujian sistem. Terdapat

juga pendekatan pengujian berdasarkan manual atau otomatis.

Pengujian otomatis dapat meminimalisir apa yang tidak bisa di-*back-up* oleh pengujian secara manual atau menghindari *human error*. Testing otomatis adalah cara yang baik untuk menghemat biaya dan waktu [6]. Terdapat banyak tools untuk pengujian otomatis dan ditujukan untuk teknik dan level dari pengujian itu sendiri. Penelitian ini ditujukan untuk memperlihatkan hasil pengujian otomatis dengan *testing tools* untuk aplikasi iPosyandu, khususnya untuk aplikasi berbasis web. Tujuan akhirnya adalah untuk *quality assurance* aplikasi iPosyandu.

Tulisan ini terdiri dari 5 bagian yaitu pendahuluan, kajian pustaka, metode, hasil dan diskusi, kesimpulan dan rencana tindak lanjut.

II. KAJIAN PUSTAKA

Berikut ini adalah kajian Pustaka yang digunakan dalam penelitian.

A. Testing

Definisi standar pengujian menurut ANSI/IEEE 1059 bahwa pengujian adalah proses menganalisis item perangkat lunak untuk mendeteksi perbedaan antara yang ada dan kondisi yang diperlukan (yaitu cacat/*errors*/*bug*s) dan untuk mengevaluasi fitur item perangkat lunak [7]. Pengujian lebih dari sekadar *debugging*. Pengujian tidak hanya digunakan untuk menemu-kan dan memperbaiki cacat. Tetapi juga digunakan dalam validasi, proses verifikasi dan pengukuran reliabilitas [6]. Pengujian merupakan kegiatan yang dilakukan untuk mengevaluasi kualitas perangkat lunak dan untuk meningkatkannya [5]. Pengujian secara manual dalam banyak kasus, memakan waktu, mahal dan kacau [8]. Beberapa alasan mengenai perkembangan otomatisasi testing adalah efisiensi pengujian lebih tinggi, akurasi dan keandalan yang lebih besar, kegunaan dan pengulangan skrip uji, cakupan tes yang ditingkatkan, simulasi lingkungan pengguna, ROI lebih tinggi: menghemat waktu dan biaya, volume dan simultanitas, dan deteksi awal bug [8]. Berikut perbedaan pengujian secara manual dan pengujian otomatis [9]. Berikut ini tabel 1 berisi rangkumgan perbedaan pengujian manual dan otomatis.

TABEL I
PERBEDAAN PENGUJIAN SECARA MANUAL DAN PENGUJIAN OTOMATIS

Pengujian Manual	Pengujian Otomatis
Investasi besar dalam sumber daya manusia	Lebih sedikit investasi dalam sumber daya manusia
Membutuhkan waktu yang banyak.	Lebih cepat
Dalam menjalankan pengujian membutuhkan beberapa kali kasus uji	Untuk menjalankan pengujian beberapa kali.
Memungkinkan penguji untuk melakukan ad-hoc tambahan (pengujian acak)	Membantu "*Compatibility Testing*" - menguji kode pada konfigurasi yang sama sekali berbeda
Bisa saja terjadi *Human Error*	Tes otomatisasi menjalankan operasi yang persis sama saat dijalankan.
Membutuhkan banyak modul pengujian untuk mencari bug pengguna nyata	Mampu menemukan hanya cacat yang diharapkan
Biaya jangka pendek berkurang.	Pengurangan Biaya Jangka Panjang

B. Web Testing

Web Testing adalah teknik yang efektif untuk memastikan kualitas aplikasi Web. Meskipun demikian, teori dan metodologi tes tradisional sulit diterapkan karena kekhususan dan kompleksitas aplikasi Web [10]. Menguji aplikasi Web tidak hanya untuk melakukan tugas-tugas penting saja, tetapi juga untuk faktor-faktor seperti keamanan, skalabilitas, dan keter-gantungan tinggi pada keluaran web browser dan cara pengguna berinteraksi web browser [10].

Beberapa tipe web testing termasuk pengujian fungsionalitas, pengujian beban dan *stress*, *browser rendering* dan pengujian kegunaan [11]. Pengujian fungsionalitas digunakan untuk me-nguji komponen web, memastikan bahwa situs web menjalankan fungsi yang diharapkan [11].

Metode *Web Testing* sebagai berikut *Functional Testing*, *Interface Testing*, *Compatibility Testing*, *Performance Testing*, dan *Security Testing* [12]. *Functional Testing* terdiri Uji semua link di halaman web, koneksi database, *form* yang digunakan di halaman web untuk mengirimkan atau mendapatkan informasi dari pengguna, pengujian Cookie. *Usability testing* yaituTes untuk navigasi: bagaimana pengguna menjelajahi halaman web, berbagai kontrol seperti tombol, kotak atau bagaimana pengguna menggunakan link pada halaman web. *Interface Testing* yaitu untuk pengujian antarmuka. Aspek pengujian *Compatibility Testing* yaitu termasuk kompatibilitas browser, kompatibilitas sistem operasi, penjelajahan seluler dan opsi pencetakan. *Performance Testing* merupakan pengujian kinerja web. *Security Testing* merupakan pengujian keamanan web.

C. Testing Tools

Alat otomatisasi pengujian perangkat lunak dapat dibagi menjadi beberapa kategori sebagai berikut : *Unit Testing Tools*, *Functional Testing Tools*, *Code Coverage Tools*, *Test Management Tools*, dan *Performance Testing Tools* [16]. Semua alat otomasi memiliki kompetensi untuk pengujian *backend* dan *frontend* pada titik yang berbeda. Selenium, Katalon, Protractor, Cucumber dan Telerik adalah alat otomatisasi yang populer[8]. Tabel 2 berikut ini berisi rangkuman fitur pada berbagai kakas pengujian otomatis.

TABEL III
FITUR *AUTOMATION TESTING TOOLS* [2]

Tool	a	b	c	d	e	f	g	h	i	j	k	l	m	n
Selenium	✓	✓	✓			✓			✓		✓			
Protractor		✓	✓						✓		✓			
UFT		✓			✓				✓					
Appium		✓	✓				✓				✓			
Test Complete					✓	✓	✓	✓						✓
Cucumber	✓	✓					✓		✓	✓	✓			
Ranorex	✓			✓			✓				✓			
Watir											✓			
RFT		✓							✓		✓			
Tricentis Tosca												✓		
Telerik Test Studio		✓	✓								✓			
Katalon Studio	✓	✓	✓	✓					✓		✓			
Linux Desktop Testing Project		✓	✓											
Serenity														
LeanFT		✓	✓											
Phantom JS		✓	✓						✓					
Coded UI		✓	✓				✓				✓			
Sikuli							✓				✓			
Applitools					✓				✓					
Test Architect									✓				✓	
Testlm									✓					

Legend: a) Record and Reply; b) Multi Programming Language Support and Codding; c) Multi Operating System Support; d) Parallel Test Running; e) AI Based Object Recognition; f) Framework Support; g)Mobile Machine & Desktop Application Test; h) Debug Full-Page Screen Shots; i) Continuous Integrity; j) Reusability; k) REST & GUI Test; l)Risk Based Testing; m)Requirement Based Testing; n) Image Based Testing

Analisis rinci harus didasarkan pada persyaratan dan kriteria tertentu dilakukan sebelum memilih alat pengujian otomatis [13]. Pemilihan alat bergantung pada berbagai faktor seperti, perangkat lunak dan tumpukan teknologinya yang akan diuji, persyaratan pengujian terperinci, keterampilan dan biaya lisensi alat. Efektif dan suksesnya pengujian otomatis tidak dapat dicapai tanpa perangkat dan kerangka kerja otomatisasi yang tepat [14].

Berdasarkan tabel perbandingan fitur *automation testing tool*, maka dalam penelitian ini menggunakan Katalon Studio dengan beberapa fitur yang dapat menunjang penelitian diantaranya record and reply, mendukung beberapa macam Bahasa pemrograman, mendukung beberapa macam system operasi, dapat menjalankan beberapa test parallel, REST dan GUI *test*.

D. Katalon

Katalon telah memperkenalkan pengujian paralel, sebuah fitur unik, di antara alat otomasi lainnya. Alat uji Selenium dan Katalon memberikan fitur paling banyak dan kemudahan penggunaan di antara alat-alat pengujian otomatis yang dibandingkan. Katalon Studio menggunakan Apache Groovy. Katalon Studio mendukung non-GUI mode untuk integrasi CI/ CD. Katalon Studio tidak mendukung pengujian terdistribusi. Katalon Studio hanya mendukung pengujian otomatisasi Web, Seluler, dan API. Katalon menyediakan Katalon Studio and Recorder untuk uji otomasi. Katalon menyediakan fitur untuk ekspor ke C#, Java, Ruby, Python, Groovy, atau Robot Framework [15].

Berikut kelebihan dan batasan Katalon Studio terangkum pada tabel 3 berikut ini:

TABEL IIIII
KELEBIHAN DAN BATASAN KATALON STUDIO

Kelebihan	Batasan
Tidak diperlukan biaya perizinan dan pemeliharaan (layanan dukungan khusus berbayar tersedia jika diperlukan).	Dukungan komunitas yang kurang
Mengintegrasikan framework dan fitur yang diperlukan untuk pembuatan dan eksekusi kasus uji cepat.	Set fitur masih berkembang.
Dibangun di atas kerangka Selenium tetapi menghilangkan kebutuhan akan keterampilan pemrograman tingkat lanjut yang diperlukan untuk Selenium.	Kurangnya pilihan untuk bahasa skrip: hanya Java / Groovy yang didukung.

Pengujian aplikasi menggunakan metode *black-box* dan alat penguji otomatis yang digunakan pada penelitian ini adalah Katalon Studio. Adapun beberapa penelitian yang juga menggunakan Katalon Studio untuk pengujian *black-box* terangkum pada tabel 4 berikut ini:

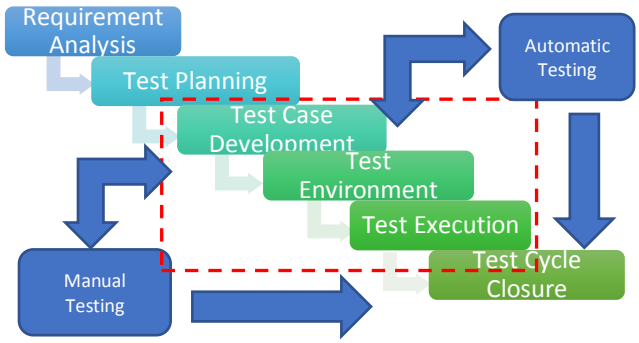
TABEL IVV
PENELITIAN TERDAHULU

Penulis	Judul	Hasil Penelitian
Ardi, F., & Putro, H. P. (2021) [16]	Pengujian Black Box Aplikasi Mobile Menggunakan Katalon Studio (Studi Kasus: ACC Partner PT. Astra Sedaya Finance)	Hasil pengujian automasi lebih efektif dibandingkan dengan pengujian secara manual yang memerlukan banyak waktu. *Tester* menjadi lebih mudah untuk mendapatkan *report* hasil pengujian karena terdapat Katalon TestOps yang menyajikan data hasil pengujian yang lebih detail dan dapat diunduh [16].
Muhtadi, M. M., Friyadi, M. D., & Rahmani, A. (2019, August)	Analisis GUI Testing pada Aplikasi E-Commerce menggunakan Katalon	Katalon sangat sensitif terhadap *error* pada *test case*. Terkadang dalam merekam aktivitas untuk test *case* banyak *event* yang dilakukan oleh pengguna aplikasi yang tidak terjalankan dalam *test case* ataupun banyak bagian-bagian yang terekam oleh sendirinya. Maka *test case* yang ada perlu diperbaiki secara manual

Penulis	Judul	Hasil Penelitian
		agar test case berjalan dengan semestinya [17]
Kosasih, Y., & Cahyono, A. B. (2021)	Automation Testing Tool Dalam Pengujian Aplikasi The Point Of Sale	Jika membutuhkan pengujian yang repetitif atau harus menguji banyak platform data yang besar sebaiknya itu di buat automation script dan jika sifatnya cuma 1 kali testing atau dibutuhkan perasaan atau eksperience langsung biasanya sifatnya emergency sebaiknya manual juga bisa dipertimbangkan dan sangat disarankan menggunakan ram 8gb dikarenakan berdasarkan pengalaman pada mengunakan ram 4gb saat melakukan playback katalon akan stuck dan pengujian yang sedang berlangsung selalu menunjukan hasil fail/gagal, setelah di upgrade ram menjadi 8gb sejauh ini katalon tidak mengalami kendala sama sekali [18].
Lee, S. J., Chen, Y. X., Ma, S. P., & Lee, W. T. (2018, July)	Test command auto-wait mechanisms for record and playback-style web application testing	Hasil percobaan menunjukkan bahwa waktu tunggu dapat ditentukan secara dinamis dan otomatis, sehingga penguji tidak perlu menambahkan secara manual perintah menunggu. Mekanisme yang diusulkan adalah diterapkan sebagai bagian dari perangkat lunak open source web testing SideeX (http://sideex.org) dan telah diadopsi lebih lanjut oleh IDE Selenium baru dan Katalon Recorder. Saat ini, lebih dari 60.000 penguji web di seluruh dunia menjalankan mekanisme tersebut untuk pengujian aplikasi web dunia [19].
Wahyuji, A. T. (2021)	Pengembangan dan Pengujian Aplikasi Website Career Center ITERA.	Pengujian menggunakan teknik white box dengan tool Katalon Studio, objek yang diujikan sesuai dengan rancangan aktivitas aplikasi. Aplikasi web career center ITERA ketika diuji membutuhkan respone time total 159.4 detik atau rata-rata 14.5 detik dengan hasil yang baik, hanya terdapat 1 kegagalan yaitu tidak bisa menerima masukkan tanggal sehingga aplikasi terdapat kesalahan. Katalon Studio tidak dapat mendeteksi masukkan berupa datepicker pada aplikasi web [20].

III. METODE PENELITIAN

Penelitian ini menggunakan metodologi yang mengadopsi siklus hidup pengujian perangkat lunak (STLC), yang dapat dilihat pada Gambar 1 berikut ini. STLC terdiri dari aktivitas *Requirement Analysis*, *Test Planning*, *Test Case Development*, *Test Environment*, *Test Execution*, *Test Cycle Closure*. Luaran *Requirement Analysis* berupa RTM (*Requirement Traceability Matrix*) dan Laporan kelayakan otomatisasi. Sedangkan untuk kegiatan *Test Planning* ditetapkan dokumen rencana pengujian /strategi dan dokumen estimasi usaha. *Test case* dan *test data* disiapkan pada aktivitas *Test Case Development*. Sedangkan pada tahapan *Test Environment* disiapkan Lingkungan siap dengan penyiapan data pengujian dan hasil *smoke test* (pengujian untuk fungsionalitas utama). Eksekusi testing dilakukan pada tahapan *Test Execution* dan ditutup dengan *Test Closure*. Kegiatan *test case developmenet*, *test environment*, dan *test execution* dilakukan dua tahap: 1) secara manual dan 2) secara otomatis. Pada kegiatan *Test Closure* dilakukan perbandingan hasil keduanya. Berikut Gambar 1 untuk metode penelitian



Gambar 1 Metode penelitian

Pada bagian ini akan dijelaskan mengenai sistem yang akan diuji serta test case yang akan diujikan, persiapan lingkungan pengujian, dan eskeskusi.

A. Gambaran. Umum Sistem iPosyandu

Sebelum melakukan pengujian, akan dijelaskan terlebih dahulu untuk aplikasi iPosyandu. Pada tahun 2017, sebuah *Action research* melanjutkan projek di KKN Unpad yang diawali dengan penelitian kualitatif dan dilanjutkan kuantitatif telah dilakukan kepada para kader posyandu dan ibu-ibu (sesuai li-terasinya). Hal tersebut dilakukan untuk mengembangkan inovasi teknologi tepat guna (TTG)/teknologi dengan tingkat ke-siapan teknologi kader. Sebuah perangkat lunak berbasis android dibuat untuk pencatatan dan pelaporan bulanan dan tahunan Sistem Informasi Posyandu (SIP) yang semula adalah berupa buku besar yang ditulis manual [1]. Aplikasi desain mock-up dibuat berdasarkan pola aktivitas kerja kader, masukan desain, dan cara pemakaian oleh kader juga inisiasi kemandirian Ibu-ibu dalam pemakaian android. Aplikasi ini di-sertai buku panduan untuk kader dan kepuasan penggunaan aplikasi iPosyandu di tahun 2018 dan diperbaharui di 2019 sesuai updating aplikasi untuk mendukung kader secara efektif dalam pelatihan menggunakan aplikasi ini. Saat ini di 2019 sedang dilakukan beta testing aplikasi oleh kader dalam menjalankan langsung posyandu di Kecamatan Pasawahan Purwakarta dan akan diperluas di beberapa daerah lainnya di Indonesia selain Purwakarta, seperti daerah binaan PT Astra International tbk yang tersebar di seluruh Indonesia. Aplikasi ini pun sedang ditambahkan menu agar kader dapat meng-input data wanita usia subur (WUS) dan Pasangan Usia Subur (PUS). Gambar 2 adalah gambaran umum sistem aplikasi iPosyandu.

Terdapat dua aplikasi, aplikasi mobile yang digunakan untuk para kader (https://play.google.com/store/apps/details?id=com. ltheordinary.iposyandu&hl=in&gl= US) dan aplikasi website yang digunakan oleh pihak lain sebagai verifikator dan pelaporan data (https://admin.iposyandu.id/). Berikut gambaran aplikasi iPosyandu terangkum pada Gambar 2 berikut ini:

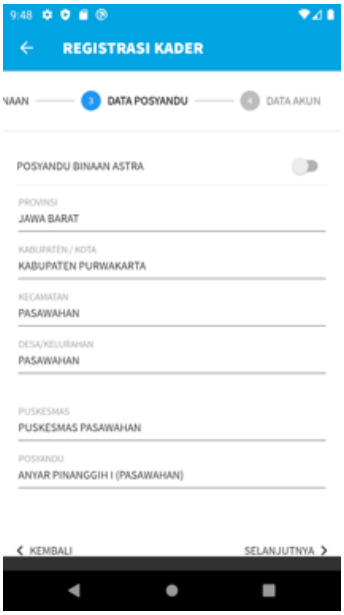
Gambar 2 Gambaran umum iPosyandu

Berikut Gambar 3, Gambar 4, Gambar 5, dan Gambar 6 yang memperlihatkan gambaran antarmuka aplikasi iPosyandu pada aplikasi mobile:

1. Halaman Pendaftaran Kader. Kader dapat mendaftar dengan mengisi semua isian di halaman ini. Di halaman ini juga terdapat ketentuan penggunaan aplikasi yang harus disetujui sebelum mendaftar.

Gambar 3 Halaman pendaftaran

2. Halaman Login. Kader yang telah mendaftar dapat masuk ke aplikasi melalui halaman ini dengan memasukkan nomor handphone dan password yang telah dibuat.



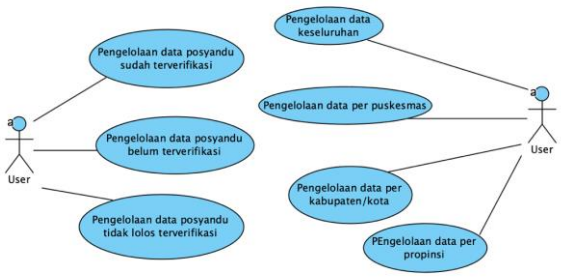
Gambar 4 Halaman pendaftaran (lanjutan)

3. Halaman Utama. Pada halaman ini terdapat ringkasan informasi mengenai jumlah bayi dan balita terdaftar, ibu hamil terdaftar, dan wanita usia subur/pasangan usia subur terdaftar. Selain itu terdapat juga informasi mengenai form evaluasi kepuasan kader terhadap aplikasi dan statistic kegiatan posyandu pada tahun tersebut.

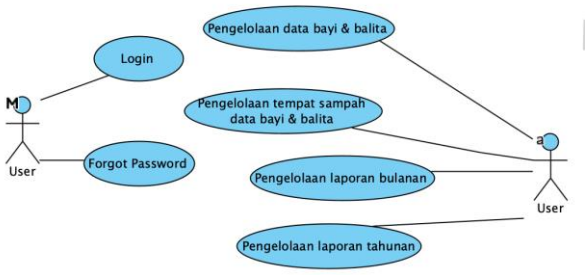
Gambar 5 Halaman *login* kader

Gambar 6 Halaman *dashboard* kader

Sedangkan fungsionalitas aplikasi iPosyandu berbasis web, dapat dilihat pada Diagram Use Case pada Gambar 7 dan Gambar 8 berikut ini:



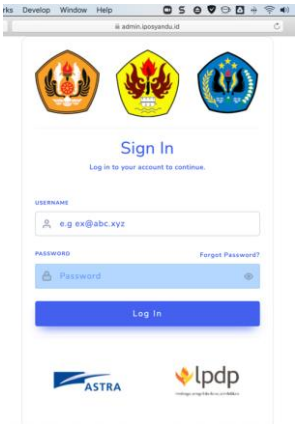
Gambar 7 Diagram *use-case* iPosyandu



Gambar 8 Diagram *use-case* iPosyandu (2)

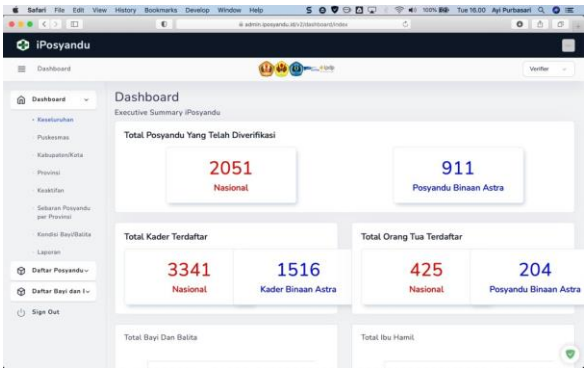
Berikut gambaran antarmuka aplikasi iPosyandu pada aplikasi berbasis web. Aplikasi ini mendapatkan data input dari aplikasi mobile. Aplikasi berbasis web ini yang akan difokuskan untuk pengujian para penelitian ini. Hal ini dikarenakan aplikasi web iPosyandu ditujukan untuk banyak *stakeholder* dari berbagai kalangan. Berikut beberapa *user interface* aplikasi iPosyandu berbasis web dapat dilihat pada Gambar 9, Gambar 10, Gambar 11, Gambar 12, Gambar 13, Gambar 14, Gambar 15, Gambar 16, dan Gambar 17.

1. Halaman Login Pengguna. Pengguna terdiri dari berbagai pihak yang diperbolehkan mengkases aplikasi, antara lain bidan koordinator, desa, dinas kesehatan, petugas gizi/pkm dan lain-lain.



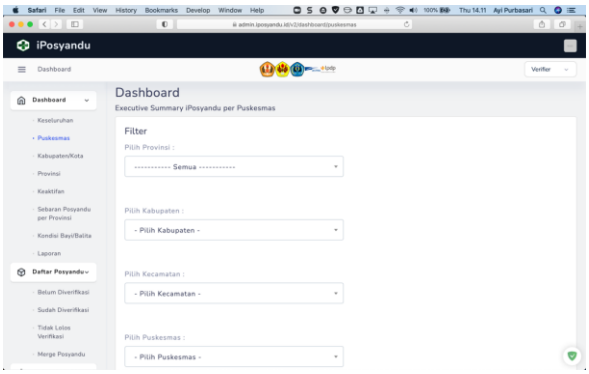
Gambar 9 Halaman *login* pengguna (*web*)

2. Halaman *Dashboard*. Halaman ini berisi rangkuman data terkait Posyandu yang terverifikasi, kader dan orang tua yang terdaftar dan data lainnya.

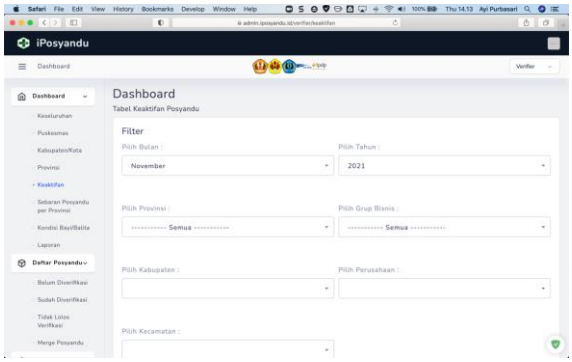


Gambar 10 Halaman *dashboard* pengguna (*web*)

3. Halaman *Dashboard* yang dilengkapi filter. Terdapat beberapa fasilitasi filter untuk data pada dashboard ini.

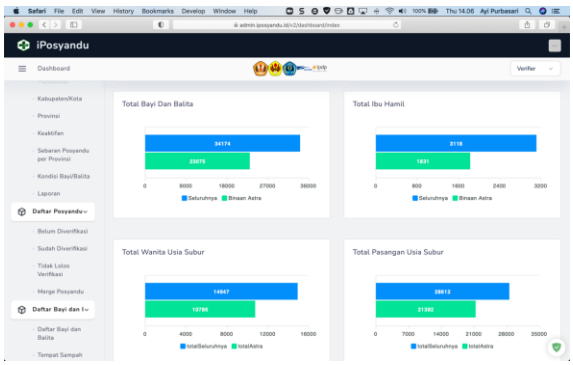


Gambar 11 Halaman filter pada *dashboard*

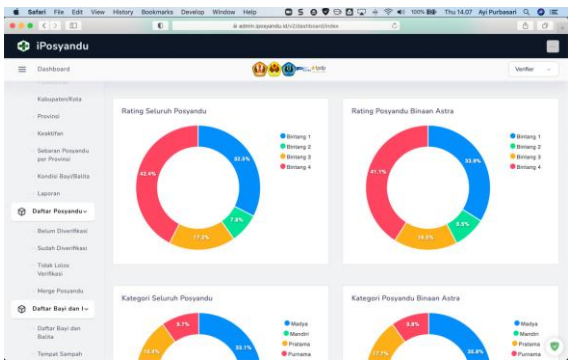


Gambar 12 Halaman filter pada *dashboard*

4. Halaman *Dashboard* dalam bentuk grafik data. Terdapat beberapa fasilitasi filter untuk data pada dashboard ini.

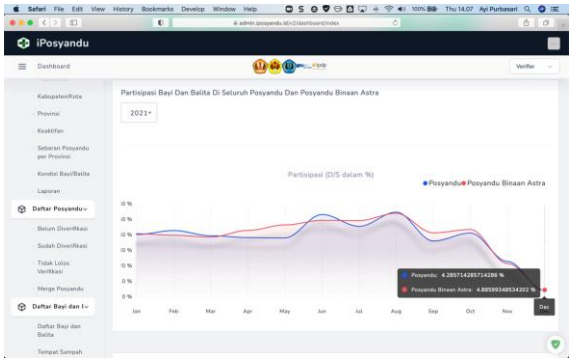


Gambar 13 Halaman *dashboard* rekapitulasi data bayi

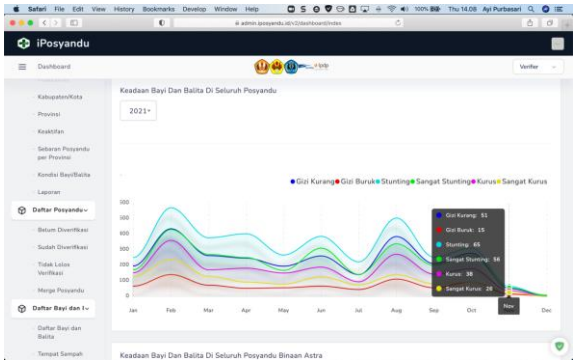


Gambar 14 Halaman *dashboard* rekapitulasi data posyandu

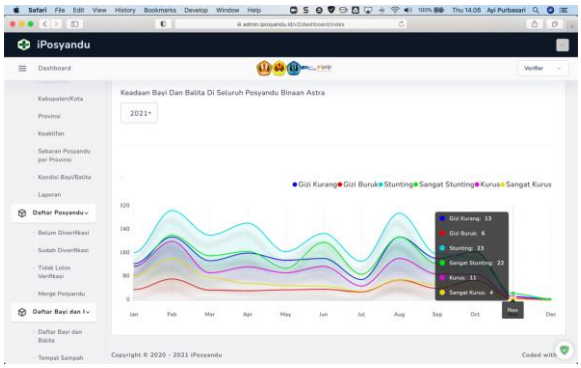
5. Halaman *Dashboard* dalam bentuk grafik untuk data utama, yaitu data Bayi dan Balita. Data ini sangat diperlukan, terutama untuk pemantauan stunting.



Gambar 15 Halaman *dashboard* grafik keadaan bayi

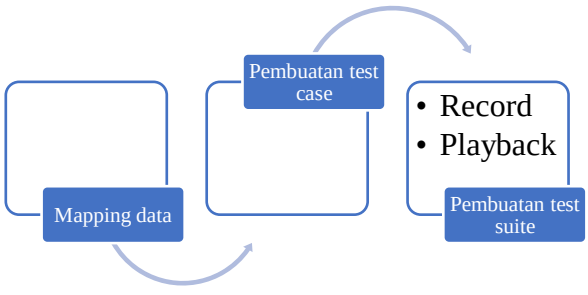


Gambar 16 Halaman *dashboard* grafik rekapitulasi data bayi



Gambar 17 Halaman *dashboard* grafik keadaan bayi

Berikut ini Gambar 18 merupakan gambaran tahapan dari Otomatisasi Testing dengan menggunakan tools Katalon, terdiri dari:



Gambar 18 Tahapan otomatisasi pengujian

A. *Mapping data*

Membuat skenario test case terlebih dahulu secara manual

B. *Pembuatan Test Case*

Berikut daftar fungsionalitas dan test case yang untuk pengujian terangkum pada tabel 5 berikut ini:

TABEL V
DAFTAR *TEST CASE*

Fungsionalitas	Test Case	Kode
Login	Kasus uji Login	TC-01
Forgot password	Kasus uji Forgot password	TC-02
Pengelolaan data posyandu – belum diverifikasi	Kasus uji data posyandu – belum diverifikasi	TC-03
Pengelolaan data posyandu sudah diverifikasi	Kasus uji data posyandu sudah diverifikasi	TC-04
Pengelolaan data posyandu tidak lolos verifikasi	Kasus uji data posyandu tidak lolos verifikasi	TC-05
Pegnelolaan data bayi dan balita	Kasus uji data bayi dan balita	TC-06
Pengelolaan data tempat sampah bayi dan balita	Kasus uji tempat sampah data bayi dan balita	TC-07
Pengelolaan data Kesseluruhan	Kasus uji data Keseluruhan	TC-08
Pengelolaan data per daftar puskesmas	Kasus uji data per daftar puskesmas	TC-09
Pengelolaan data per daftar kabupaten/kota	Kasus uji data per daftar kabupaten/kota	TC-10
Pengelolaan data per daftar provinsi	Kasus uji data per daftar provinsi	TC-11
Pengelolaan Laporan bulanan	Kasus uji Login	TC-12
Pengelolaan Laporan tahunan	Kasus uji Forgot password	TC-13

Pada pengujian automasi menggunakan katalon studio, Langkah yang dilakukan yaitu membuat test case dengan cara melakukan pengambilan object. Pengambilan object dapat dilakukan dengan *spy* dan *record object*. Perbedaan keduanya adalah *object spy* melakukan pengambilan *object* satu persatu sesuai pilihan. Sedangkan record object adalah melakukan simulasi pengujian [16].

C. *Pembuatan Test Suite*

Test suite berisi *test case* dan data files yang telah dibuat sebelumnya. Proses pengujian dimulai dengan menjalankan *test suite* dan hasil dapat dilihat pada bagian *log viewer*. Selain dari *log viewer* dapat juga dilihat melalui Katalon TestOps.

Pengujian yang dilakukan otomatis menerapkan metode record & playback. Record adalah proses perekaman semua event yang diterima dari aktivitas pengguna pada sebuah halaman web. Semua hasil perekaman akan menjadi test yang disimpan oleh Katalon, dan menjadikannya sebagai test case. Test case dapat diubah/dimodifikasi oleh penguji, untuk penyempurnaan pro-ses pengujian, jika diperlukan. Test case tersebut selanjutnya dijalankan secara otomatis menggunakan fungsi playback. Ketika test case dijalankan, log dari setiap langkah secara oto-matis akan tersimpan. Log tersebut berisi status berhasil atau gagalnya suatu langkah dan waktu yang dibutuhkan untuk menyelesaikan langkah tersebut. Ketika terdapat langkah yang gagal maka proses menjalankan test case akan berhenti dan dapat dilihat rincian dari test case tersebut pada log viewer.

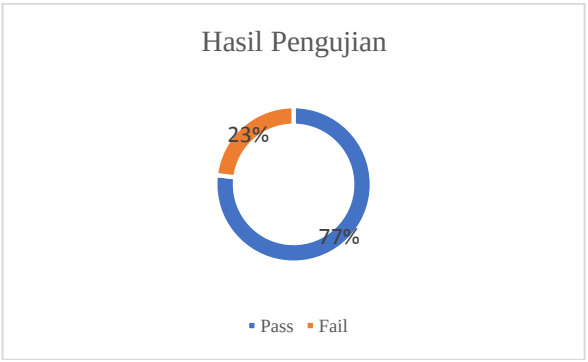
D. *Hasil Pengujian*

Hasil pengujian dari pengujian manual dan pengujian otomatis akan dibandingkan dengan parameter: status hasil eksekusi dan waktu eksekusi. Hasil pengujian dirangkum pada tabel 6 berikut ini:

TABEL VI
HASIL PENGUJIAN

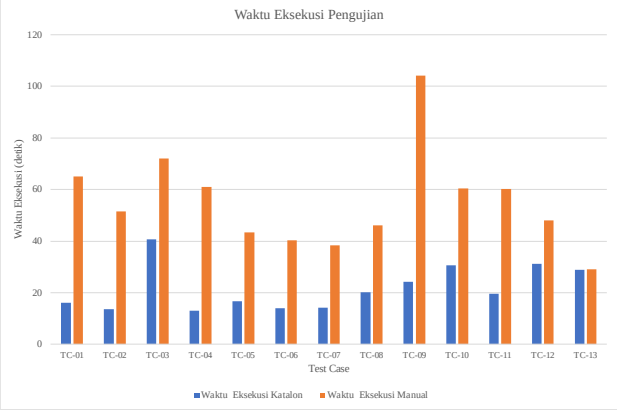
No	Test Case	Hasil eksekusi		Waktu Eksekusi (detik)	
		Katalon	Manual	Katalon	Manual
1	TC-01	Pass	Pass	16,16	65,12
2	TC-02	Pass	Pass	13,61	51,54
3	TC-03	Pass	Pass	40,72	72,02
4	TC-04	Fail	Fail	13,05	60,91
5	TC-05	Fail	Fail	16,75	43,3
6	TC-06	Pass	Pass	13,97	40,37
7	TC-07	Pass	Pass	14,2	38,36
8	TC-08	Pass	Pass	20,12	46,01
9	TC-09	Pass	Pass	24,16	104,04
10	TC-10	Pass	Pass	30,67	60,32
11	TC-11	Fail	Fail	19,63	60,19
12	TC-12	Pass	Pass	31,22	47,96
13	TC-13	Pass	Pass	28,82	29,13
Total durasi waktu pengujian				283,08	719,27

Secara hasil eksekusi, pengujian manual dan otomatis menghasilkan status yang sama: 10 pass dan 3 fail. Berikut Gambar 19 dan Gambar 20 yang merupakan grafik yang menunjukkan hasil eksekusi pengujian:



Gambar 19 Grafik hasil pengujian

Pada saat menjalankan pengujian dapat diketahui *respon time* setiap test *case* yang. *Respon time* pada satu *test case* berbeda-beda apabila dijalankan beberapa kali. Total waktu eksekusi pengujian otomatis total adalah 283,08 detik, dan 719,27 detik untuk pengujian secara manual. Dengan demikian terjadi peningkatan kecepatan 2,54 kali atau 254%.



Gambar 20 Grafik waktu pengujian

Katalon studio terdapat fitur *Spy* dan *Record object*, dimana element dapat terekam dan dapat disimpan sebagai *object*. Namun pada penelitian ini, *spy* dan *record object* pada Katalon Studio belum dimanfaatkan.

IV. KESIMPULAN

Pengujian memperlihatkan adanya fungsionalitas yang masih harus diperbaiki bersamaan dengan berbagai penambahan fungsionalitas lainnya juga. Pemanfaatan Katalon Studio sangat berguna untuk efektifitas dan meminimalisir human error ketika dilakukan pengujian secara manual. Report hasil pengujian lebih mudah didapatkan karena Katalon Studio menyediakan Katalon TestOps. Katalon TestOps menyajikan data hasil pengujian selain dengan format csv, terdapat juga format xlsx dan pdf.

Pengujian untuk aplikasi mobile perlu dilakukan, terutama aplikasi mobile ini menjadi sumber input dari aplikasi iPosyandu dan digunakan oleh kalangan kader dengan tingkat pemahaman digital yang beragam. Dengan demikian, penelitian berikutnya akan menetapkan pengujian aplikasi mobile dengan penggunaan Katalon Studio.

ACKNOWLEDGE

Penelitian ini didanai oleh RISPRO Riset Inovasi - LPDP kerjasama Universitas Padjajaran dengan Universitas Pasundan, Universitas Langlangbuana, dan Dinas Kesehatan Kabupaten Purwakarta.

REFERENSI

[1] F. R. Rinawan *et al.*, "Understanding mobile application development and implementation to monitor Posyandu data in Indonesia: a 3-year hybrid action research to build 'a bridge' from the community to the national scale," 2020.

[2] A. I. Susanti, F. R. Rinawan, and I. Amelia, "Mothers knowledge and perception of toddler growth monitoring using iPosyandu application," *Global Medical and Health Communication (GMHC)*, vol. 7, no. 2, pp. 93–99, 2019, doi: 10/gndnpm.

[3] A. I. Susanti, A. N. S. Didah, D. Ferdian, and F. R. Rinawan, "Persepsi Petugas Gizi Dalam Pemantauan Status Gizi Balita dengan Menggunakan Website iPosyandu," *JURNAL KEBIDANAN*, vol. 6, no. 3, pp. 376–382, 2020.

[4] W. Widarti, F. R. Rinawan, A. I. Susanti, and H. N. Fitri, "Perbedaan pengetahuan kader posyandu sebelum dan sesudah dilakukan pelatihan penggunaan aplikasi iPOSYANDU," *Jurnal Pengabdian dan Pengembangan Masyarakat*, vol. 1, no. 2, pp. 143–150, 2018, doi: 10/gndnwb.

[5] I. Jovanović, "Software testing methods and techniques," *The IPSI BgD Transactions on Internet Research*, vol. 30, 2006.

[6] J. Pan, "Software testing," *Dependable Embedded Systems*, vol. 5, 1999.

[7] S. K. Singh and A. Singh, *Software testing*. Vandana Publications, 2012.

[8] D. Ateşoğulları and A. Mishra, "Automation testing tools: a comparative view," *International Journal on Information Technologies & Security*, vol. 12, no. 4, 2020.

[9] N. Srivastava, U. Kumar, and P. Singh, *Software and Performance Testing Tools*. 2021.

[10] Z. Qian, H. Miao, and H. Zeng, "A practical web testing model for web application testing," in *2007 third international IEEE conference on signal-image technologies and internet-based system*, Dec. 2007, pp. 434–441.

[11] C. Kallepalli and J. Tian, "Measuring and modeling usage and reliability for statistical web testing," *IEEE transactions on software engineering*, vol. 27, no. 11, pp. 1023–1036, 2001, doi: 10/fsmhn5.

[12] S. Kundu, "Web testing: tool, challenges and methods," *IJCSI International Journal of Computer Science Issues*, vol. 9, no. 2, pp. 1694–0814, 2012.

[13] H. P. Bajaj, "Choosing the right automation tool and framework is critical to project success," *Infosys Limited*, vol. 41, 2015.

[14] M. A. Umar and C. Zhanfang, "A study of automated software testing: Automation tools and frameworks," *International Jour-nal of Computer Science Engineering (IJCSE*, vol. 6, pp. 217–225, 2019.

[15] Katalon Studio, "A Comparison of Automated Testing Tools," *A Comparison of Automated Testing Tools*. https://www.katalon.com/resourcescenter/blog/comparison-automated-testing-tools/.

[16] F. Ardi and H. P. Putro, "Pengujian Black Box Aplikasi Mobile Menggunakan Katalon Studio (Studi Kasus: ACC Partner PT," *Astra Sedaya Finance). AUTOMATA*, vol. 2, no. 1, 2021.

[17] M. M. Muhtadi, M. D. Friyadi, and A. Rahmani, "Analisis GUI testing pada aplikasi e-commerce menggunakan katalon," *Prosiding Industrial Research Workshop and National Seminar*, vol. 10, no. 1, pp. 1387–1393, Aug. 2019.

[18] Y. Kosasih and A. B. Cahyono, "Automation Testing Tool Dalam Pengujian Aplikasi The Point Of Sale," *AUTOMATA*, vol. 2, no. 1, 2021.

[19] S. J. Lee, Y. X. Chen, S. P. Ma, and W. T. Lee, "Test command auto-wait mechanisms for record and playback-style web application testing," in *2018 IEEE 42nd annual computer software and applications conference (COMPSAC*, Jul. 2018, vol. 2, pp. 75–80.

[20] A. T. Wahyuji, "Pengembangan dan Pengujian Aplikasi Web-site Career Center ITERA," *Jurnal Tekno Kompak*, vol. 15, no. 1, pp. 67–78, 2021.