

Pengujian Otomatis Website DPRD Kabupaten Bengkalis Menggunakan Data-driven Testing *Automated Testing of the Bengkalis Regency Regional House of Representatives (DPRD) Website Using a Data-Driven Testing Approach*

Arif Rahman*¹, Fajar Ratnawati²

^{1, 2}Program Studi Rekayasa Perangkat Lunak, Politeknik Negeri Bengkalis, Sungai Alam,
Bengkalis, Riau, +62822 3572 0044

E-mail : ar6908215@gmail.com*¹, fajar@polbeng.ac.id²

*Corresponding author

Received 7 March 2026; Revised 6 April 2026; Accepted 9 April 2026

Abstrak - Pengujian perangkat lunak merupakan tahap penting untuk memastikan sistem berjalan sesuai dengan kebutuhan fungsional. Penelitian ini bertujuan menerapkan metode Data-Driven Testing (DDT) untuk melakukan pengujian otomatis pada sistem admin website DPRD Kabupaten Bengkalis. Metode DDT memungkinkan pengujian dilakukan secara sistematis dengan memanfaatkan variasi data uji yang disimpan pada sumber data eksternal. Proses penelitian meliputi analisis kebutuhan, perancangan skenario pengujian, penyusunan data uji, implementasi otomatisasi menggunakan Katalon Studio, serta evaluasi hasil pengujian. Data uji disimpan dalam file Microsoft Excel dan diintegrasikan dengan skrip pengujian melalui mekanisme data binding sehingga setiap kombinasi input dapat dieksekusi secara otomatis. Form login digunakan sebagai contoh utama implementasi metode ini karena merupakan gerbang autentikasi pada sistem admin. Hasil pengujian menunjukkan bahwa mekanisme autentikasi pada form login mampu memproses berbagai kombinasi input dengan baik, dimana sistem hanya memberikan akses ketika kredensial valid dan menolak input yang tidak sesuai. Selain itu, evaluasi terhadap beberapa form lain menunjukkan masih adanya kelemahan pada mekanisme validasi input yang memungkinkan data tidak lengkap tetap tersimpan dalam sistem. Hasil penelitian ini menunjukkan bahwa penerapan Data-Driven Testing dapat meningkatkan efisiensi proses pengujian serta membantu mengidentifikasi kelemahan validasi pada sistem berbasis web.

Kata Kunci - Data-driven Testing, Pengujian Fungsional, Katalon Studio, Website DPRD

Abstract - Software testing is an essential process to ensure that a system operates according to its functional requirements. This study aims to implement the Data-Driven Testing (DDT) method to perform automated testing on the admin system of the Bengkalis Regency DPRD website. The DDT approach enables systematic testing by utilizing various test data stored in external data sources. The research process includes requirement analysis, test scenario design, test data preparation, automated test implementation using Katalon Studio, and evaluation of testing results. Test data were stored in Microsoft Excel and integrated with test scripts through a data binding mechanism, allowing each input combination to be executed automatically. The login form was used as the main case study because it functions as the authentication gateway of the admin system. The testing results indicate that the login authentication mechanism correctly handles various input combinations by granting access only when valid credentials are provided and rejecting invalid inputs. Furthermore, the evaluation of several other forms reveals weaknesses in input validation mechanisms that still allow incomplete data to be stored in the system. These findings indicate that the implementation of Data-Driven Testing can improve testing efficiency and help identify validation weaknesses in web-based systems.

Keywords - Data-driven Testing, Functional Testing, Katalon Studio, DPRD Website

1. PENDAHULUAN

Website menjadi komponen penting dalam transformasi digital pelayanan publik, termasuk pada DPRD Kabupaten Bengkalis. Sebagai media penyampaian informasi dan pengelolaan *data internal*, website dituntut memiliki kualitas fungsional yang baik, khususnya pada sisi admin. Proses *validasi input*, seperti pada fitur *login* dan pengelolaan berita, harus mampu membedakan *data valid* dan tidak *valid* agar terhindar dari kesalahan data maupun gangguan alur kerja[1]. Seiring meningkatnya kompleksitas aplikasi web, pengujian manual dinilai kurang efektif karena rentan kesalahan, memakan waktu, dan sulit dijalankan secara konsisten. Salah satu tantangan dalam pengujian perangkat lunak di instansi pemerintahan adalah belum optimalnya penerapan pengujian otomatis berbasis data yang terstruktur. Pendekatan *data-driven testing* (DDT) menawarkan solusi dengan memungkinkan pengujian berbagai skenario *input* secara sistematis dan otomatis[2]. Berbagai penelitian telah membahas pengujian perangkat lunak dari berbagai pendekatan. Maylan Anggi Puspita dkk. melakukan *User Acceptance Testing* (UAT) dan *performance testing* pada website Kanal Pengetahuan Dikti menggunakan PSSUQ dan Apache JMeter. Hasil UAT menunjukkan penilaian positif (rata-rata 2,25 dari 82 responden), namun pengujian performa dengan 40.000 pengguna menghasilkan tingkat error 59%, menandakan perlunya pengujian performa yang lebih menyeluruh pada sistem berbasis web publik[3]. Alfian Arifandi dkk. melalui metode *Systematic Literature Review* (SLR) mengidentifikasi bahwa teknik *black box* dan *white box testing* merupakan metode yang paling umum digunakan dalam praktik pengujian perangkat lunak[4]. Hal ini diperkuat oleh Ilham Rafif Dhaifullah dkk. yang menyatakan bahwa pemilihan metode (*black-box*, *white-box*, *grey-box*) dipengaruhi oleh kebutuhan proyek, kompleksitas sistem, serta pengalaman pengembang[5].

Pendekatan *black box* banyak diterapkan dalam berbagai studi. Wijaya K. menekankan pentingnya integrasi *manual* dan *automated testing* dalam QA sistem ERP untuk mempercepat validasi dan mengurangi risiko kesalahan fungsi[6]. Vanessa Angela Amei membandingkan pengujian manual dan otomatis (Katalon Studio) pada website ACC, menemukan bahwa otomatisasi meningkatkan efisiensi praktis hingga 55% meskipun membutuhkan waktu lebih lama, serta berhasil mengidentifikasi *bug* sistem[7]. Masitha Putri Ardhana Ginting juga menunjukkan bahwa *black box testing* efektif dalam mengidentifikasi kesalahan fungsional pada aplikasi berbasis *web* instansi pemerintah[8]. Namun, efektivitas metode *black box* tidak selalu optimal. Stevanu Dika Pratama menemukan nilai *Defect Removal Efficiency* (DRE) hanya 21%, menunjukkan masih banyak kesalahan yang tidak terdeteksi[9]. Mintarsih menambahkan bahwa teknik *state transition* dalam *black box testing* mampu mendeteksi *bug* yang tidak terlihat pada pengujian *input-output* statis[10]. Danendra Khansa Pallas Wahyudi menemukan 24 kegagalan dari 256 *test case black box* pada aplikasi POST, menunjukkan pentingnya perbaikan berkelanjutan meskipun sistem dinilai layak[11]. Dari sisi performa sistem, Muhamad Rafli menunjukkan bahwa strategi *load balancing* Round Robin memberikan hasil terbaik dalam menangani beban tinggi hingga 40.000 *request*[12], menegaskan pentingnya pengujian performa dalam sistem dengan akses simultan besar. Sementara itu, penelitian terkait *data-driven testing* (DDT) menunjukkan hasil yang lebih adaptif dan terstruktur. Praveen Kumar Koppanati menjelaskan bahwa penggunaan JSON dan XML dalam DDT memungkinkan pemisahan data uji dari skrip sehingga meningkatkan fleksibilitas dan skalabilitas pengujian otomatis[13]. Lucia Sri Istiyowat menunjukkan bahwa otomatisasi berbasis DDT menggunakan Katalon Studio mencapai tingkat keberhasilan 80,21% dan mempermudah implementasi *smoke* serta *regression testing*[14]. Sa'adah mengembangkan *framework* otomatis berbasis *Serenity* dan *Jenkins* yang mempercepat proses uji dan meningkatkan konsistensi pengujian[15]. Lebih lanjut, Mohnish Neelapu mengembangkan *framework* otomatisasi berbasis data yang meningkatkan cakupan pengujian, mengurangi waktu eksekusi hingga 40%, dan meningkatkan deteksi kesalahan sebesar 25% melalui pendekatan adaptif berbasis *machine learning*[16]. Andrzej Benjamin Bujok melalui kerangka MED-V STEP juga menunjukkan bahwa pendekatan pengujian terstruktur mampu

meningkatkan konsistensi serta deteksi dini cacat sistem, meskipun diterapkan pada perangkat lunak medis[17].

Temuan-temuan tersebut menunjukkan bahwa pendekatan pengujian otomatis berbasis data dan *framework* terstruktur mampu meningkatkan efektivitas serta kualitas proses pengujian perangkat lunak. Secara khusus, *data-driven testing* memiliki keunggulan dalam pemisahan skrip dan data uji, sehingga satu skrip dapat digunakan untuk berbagai variasi *input*. Hal ini menjadikan proses pengujian lebih sistematis, terdokumentasi, dan mudah ditelusuri. Meskipun demikian, penerapan metode ini masih terbatas di sektor pemerintahan daerah, khususnya dalam pengujian validasi *input* pada sistem *admin website* layanan publik. Oleh karena itu, penelitian ini menerapkan metode *data-driven testing* pada *website* DPRD Kabupaten Bengkalis untuk memastikan setiap skenario penggunaan dapat diuji secara otomatis, terstruktur, dan konsisten. Hasil pengujian diharapkan memberikan evaluasi teknis yang terukur serta kontribusi akademik dalam penerapan pengujian otomatis berbasis data di lingkungan pemerintahan.

Meskipun berbagai penelitian telah membahas pengujian perangkat lunak menggunakan metode *automated testing* maupun *data-driven testing*, sebagian besar penelitian tersebut berfokus pada sistem umum atau lingkungan industri, serta lebih menekankan pada aspek efisiensi pengujian tanpa mengkaji secara spesifik kelemahan validasi input pada sistem pemerintahan daerah. Selain itu, implementasi *Data-Driven Testing* pada sistem berbasis web di lingkungan instansi pemerintah, khususnya pada level *admin website* pelayanan publik, masih belum banyak dibahas secara mendalam.

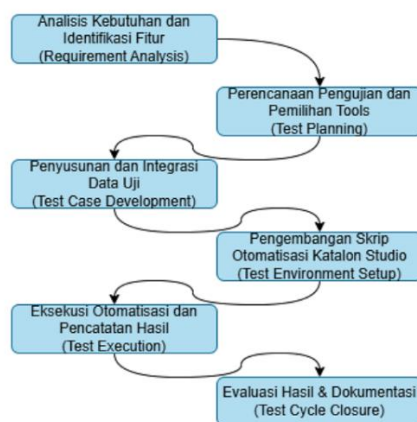
Oleh karena itu, penelitian ini memiliki kebaruan pada tiga aspek utama, yaitu:

- (1) implementasi *Data-Driven Testing* pada sistem *admin website* DPRD sebagai representasi sistem pemerintahan daerah,
- (2) identifikasi kelemahan validasi input pada berbagai modul sistem secara sistematis, dan
- (3) evaluasi efektivitas penggunaan Katalon Studio dalam mendukung pengujian otomatis berbasis data.

Dengan demikian, penelitian ini tidak hanya berfokus pada proses otomatisasi pengujian, tetapi juga memberikan analisis terhadap kualitas validasi sistem serta dampaknya terhadap integritas data.

2. METODE PENELITIAN

Metode yang digunakan pada penelitian ini menggunakan pendekatan *Data-Driven Testing* (DDT) untuk melakukan pengujian otomatis pada sistem *admin Website* DPRD Kabupaten Bengkalis. Proses penelitian disusun berdasarkan tahapan *Software Testing Life Cycle* (STLC) yang meliputi perencanaan pengujian, perancangan skenario uji, penyusunan data uji, pelaksanaan pengujian otomatis, serta evaluasi hasil pengujian. Alur prosedur penelitian yang digunakan dalam penelitian ini ditunjukkan pada Gambar 1.



Gambar 1 Prosedur Penelitian

Berdasarkan Gambar 1, penelitian dilakukan dengan beberapa prosedur yakni sebagai berikut:

1. Analisis Kebutuhan dan Identifikasi Fitur

Tahap awal penelitian dilakukan dengan menganalisis sistem yang akan diuji, yaitu *website* DPRD Kabupaten Bengkalis, dengan fokus pada fitur yang tersedia pada halaman *admin*. Proses analisis dilakukan dengan menelusuri berbagai halaman admin serta mengidentifikasi fitur yang melibatkan proses *input data* dan validasi sistem. Berdasarkan hasil analisis, terdapat beberapa fitur yang memiliki elemen form input dan berpotensi diuji menggunakan pendekatan *Data-Driven Testing*, yaitu:

- a. Form Login Admin
- b. Form Informasi Publik
- c. Form Polling
- d. Form Agenda Kegiatan
- e. Form Pengumuman
- f. Form Kepegawaian
- g. Form Berita
- h. Form Berita Foto
- i. Form Video
- j. Form Galeri
- k. Form Agenda Protokoler

Fitur-fitur tersebut dipilih karena melibatkan proses validasi *input* yang memungkinkan dilakukan pengujian dengan berbagai variasi data secara otomatis. Dalam penelitian ini, *form login* digunakan sebagai contoh utama untuk menggambarkan implementasi metode *Data-Driven Testing* karena merupakan pintu masuk sistem dan melibatkan proses autentikasi pengguna.

2. Perencanaan Pengujian dan Pemilihan *Tools*

Pada tahap ini dilakukan penyusunan strategi pengujian serta pemilihan *tools* yang digunakan dalam proses otomatisasi. Katalon Studio dipilih sebagai alat bantu pengujian karena mendukung integrasi dengan sumber *data* eksternal seperti Microsoft Excel melalui fitur *data binding*. Langkah-langkah yang dilakukan dalam tahap perencanaan pengujian meliputi:

- a. Menentukan fitur pada halaman admin yang akan diuji.
- b. Menyusun variasi data uji berupa kombinasi input valid dan tidak valid.
- c. Menyusun *test scenario* dan *test case* berdasarkan fitur yang diuji.
- d. Menyusun struktur *file Excel* sebagai sumber data pada metode *Data-Driven Testing*.

Selain itu, jumlah *test case* yang disusun pada penelitian ini disesuaikan dengan kompleksitas masing-masing form yang diuji. Pada form login, terdapat 5 skenario utama dengan total 5 kombinasi data uji. Sementara itu, pada keseluruhan modul admin yang meliputi beberapa fitur utama, jumlah *test case* yang digunakan mencapai 11 *test case* yang mencakup variasi input valid dan tidak valid.

Penentuan jumlah *test case* ini bertujuan untuk memastikan bahwa setiap kemungkinan kondisi input, baik valid maupun tidak valid, dapat diuji secara sistematis sehingga mampu mengidentifikasi potensi kesalahan pada mekanisme validasi sistem.

3. Penyusunan dan Integrasi Data Uji

Data uji disusun menggunakan Microsoft Excel yang berisi berbagai kombinasi *input* untuk setiap fitur yang diuji. Pada pendekatan *Data-Driven Testing*, setiap baris data pada file Excel merepresentasikan satu skenario pengujian yang akan dijalankan secara otomatis oleh skrip pengujian di Katalon Studio. Data uji yang digunakan dalam penelitian ini dibagi menjadi dua kategori utama, yaitu:

- a. Data *valid*, yaitu data yang sesuai dengan ketentuan sistem.
- b. Data tidak *valid*, yaitu data yang tidak memenuhi aturan validasi sistem.

Struktur file Excel yang digunakan sebagai sumber data uji terdiri dari beberapa kolom utama, yaitu *username*, *password*, *expected_result*, dan keterangan. Setiap kolom tersebut dipetakan ke dalam variabel pada skrip pengujian sehingga memungkinkan proses pengujian dilakukan secara otomatis dan berulang. Setiap baris pada file Excel merepresentasikan satu skenario pengujian

yang akan dieksekusi oleh sistem. Contoh format data uji untuk fitur login ditunjukkan pada Tabel 1.

Tabel 1 Format Data Uji Login

Username	Password
admin01	Admin123
admin02	Admin123
admin02	wrong123
usertest	usertest
-	-

Data uji tersebut kemudian dihubungkan dengan skrip otomatisasi melalui mekanisme data binding pada Katalon Studio dengan memetakan setiap kolom Excel ke variabel input seperti username dan password. Pada saat eksekusi, Katalon Studio akan membaca setiap baris data secara iteratif dan menjalankan skenario pengujian secara otomatis berdasarkan data yang tersedia.

3.1. Perancangan Test Scenario dan Test Case

Setelah *data* uji disusun, tahap berikutnya adalah merancang *test scenario* dan *test case* untuk setiap fitur yang diuji. *Test scenario* digunakan untuk menggambarkan kondisi pengujian secara umum, sedangkan *test case* digunakan untuk mendefinisikan langkah pengujian secara lebih spesifik. Contoh skenario pengujian pada *form login* ditunjukkan pada Tabel 2.

Tabel 2 Test Scenario Form Login

Nama Modul / ID Persyaratan	Nama Skenario Uji	ID kasus uji	Nama kasus uji
Form Login	Verifikasi Pengisian <i>Form</i> Login	TC_LOGIN_01	Login dengan username dan password valid
		TC_LOGIN_02	Login dengan username salah dan password benar
		TC_LOGIN_03	Login dengan username benar dan password salah
		TC_LOGIN_04	Login dengan username salah dan password salah
		TC_LOGIN_05	Login dengan username dan password kosong

Skenario pengujian ini digunakan untuk memverifikasi apakah sistem *login* mampu melakukan validasi terhadap berbagai kombinasi input serta memastikan bahwa hanya pengguna dengan kredensial yang valid yang dapat mengakses halaman *admin*.

Test case disusun berdasarkan skenario pengujian yang telah dirancang sebelumnya untuk memverifikasi mekanisme autentikasi pada sistem *admin*. Setiap *test case* mendefinisikan kondisi pengujian, data uji, langkah pengujian, serta hasil yang diharapkan. Pengujian dilakukan menggunakan pendekatan *Data-Driven Testing*, sehingga setiap kombinasi data uji dapat dieksekusi secara otomatis menggunakan skrip pengujian pada Katalon Studio. Contoh *test case* untuk pengujian *form login admin* ditunjukkan pada Tabel 3.

Tabel 3 Test Case Form Login

Skenario Kasus Uji	Nama Kasus Uji	Prasyarat	Data Uji	Langkah Pengujian	Hasil yang Diharapkan
Cek login dengan akun valid	Login dengan username dan password valid	Form login tersedia	Username: admin Password: admin123	1. Buka halaman login 2. Masukkan username dan password valid 3. Klik tombol login	Sistem mengarahkan ke dashboard admin
Cek validasi username salah	Login dengan username salah dan password benar	Form login tersedia	Username: adminn Password: admin123	1. Buka halaman login 2. Masukkan password salah 3. Klik tombol login	Sistem menampilkan pesan "Username tidak ditemukan"
Cek validasi password salah	Login dengan username benar dan password salah	Form login tersedia	Username: admin Password: salahpass	1. Buka halaman login 2. Masukkan password salah 3. Klik tombol login	Sistem menampilkan pesan "Password salah"
Cek validasi username dan password salah	Login dengan username dan password kosong	Form login tersedia	Username: coba Password: coba124	1. Buka halaman login 2. Masukkan username dan password salah 3. Klik tombol login	Sistem menampilkan pesan "Username dan Password salah"
Cek validasi field kosong	Login dengan username dan password kosong	Form login tersedia	Username: - Password: -	1. Buka halaman login 2. Biarkan kolom kosong 3. Klik tombol login	Sistem menampilkan pesan "Semua field wajib diisi"

Pengujian ini bertujuan untuk memastikan bahwa sistem *login* mampu melakukan validasi terhadap berbagai kombinasi input yang dimasukkan oleh pengguna. Melalui pendekatan *Data-Driven Testing*, setiap kombinasi data uji dapat dijalankan secara otomatis sehingga proses pengujian menjadi lebih sistematis, konsisten, dan mudah direplikasi. Hasil dari pengujian ini digunakan untuk mengevaluasi apakah sistem mampu membedakan antara kredensial yang valid dan tidak valid serta memastikan keamanan akses menuju halaman admin.

3.2. Implementasi *Data-Driven Testing*

Implementasi metode *Data-Driven Testing* pada penelitian ini dilakukan dengan mengintegrasikan data uji yang tersimpan dalam file Microsoft Excel dengan skrip pengujian otomatis pada Katalon Studio. Melalui pendekatan ini, setiap baris data dalam *file Excel* diperlakukan sebagai satu skenario pengujian yang akan dieksekusi secara otomatis oleh sistem.

a. Integrasi Data Uji

File Excel digunakan sebagai sumber data uji yang berisi berbagai kombinasi input username dan password. Data tersebut kemudian dihubungkan dengan skrip pengujian melalui fitur data binding pada Katalon Studio. Dengan mekanisme ini, skrip pengujian dapat membaca setiap baris data secara otomatis dan menjalankan proses pengujian secara berulang tanpa perlu mengubah skrip utama. Pendekatan ini memungkinkan proses pengujian dilakukan secara lebih fleksibel karena variasi *data* uji dapat ditambahkan atau dimodifikasi langsung pada file Excel tanpa memerlukan perubahan pada skrip pengujian.

b. Konfigurasi Objek Halaman

Tahap selanjutnya adalah melakukan konfigurasi objek halaman yang akan digunakan dalam proses pengujian otomatis. Objek halaman dikelola melalui Object Repository pada Katalon Studio dan dipetakan berdasarkan atribut *HTML* atau *XPath* dari elemen yang terdapat pada halaman *web*. Objek yang digunakan pada pengujian form login meliputi:

- a) *Field Username*, yaitu elemen *input* yang digunakan untuk memasukkan nama pengguna.
- b) *Field Password*, yaitu elemen *input* yang digunakan untuk memasukkan kata sandi.
- c) Tombol Login, yaitu elemen tombol yang digunakan untuk mengeksekusi proses autentikasi.

Setiap objek tersebut dikonfigurasi agar dapat dipanggil secara otomatis oleh skrip pengujian selama proses eksekusi berlangsung.

c. Verifikasi Hasil Pengujian

Proses verifikasi dilakukan untuk menentukan apakah skenario *login* berhasil atau gagal. Dalam penelitian ini, keberhasilan *login* ditentukan berdasarkan perubahan halaman setelah proses autentikasi dijalankan.

- a) *Login* berhasil ditandai dengan berpindahannya halaman dari *form login* ke halaman dashboard admin.
- b) *Login* gagal ditandai dengan tetap berada pada halaman *login* setelah tombol *login* ditekan.

Selain berdasarkan perpindahan halaman, proses verifikasi juga dilakukan dengan memeriksa keberadaan elemen tertentu pada halaman dashboard menggunakan fungsi verifikasi elemen pada Katalon Studio. Pendekatan ini digunakan untuk memastikan bahwa hasil pengujian lebih akurat dan tidak hanya bergantung pada perubahan halaman.

4. Pengembangan Skrip Otomatisasi

Pada tahap ini dilakukan pengembangan skrip pengujian otomatis menggunakan Katalon Studio. Skrip tersebut dirancang untuk membaca setiap baris *data* dari *file Excel* dan mengeksekusi proses *login* secara otomatis menggunakan kombinasi *username* dan *password* yang telah disiapkan. Dengan memanfaatkan pendekatan *Data-Driven Testing*, skrip pengujian dapat digunakan untuk menjalankan berbagai skenario pengujian tanpa perlu melakukan perubahan pada kode utama. Proses ini memungkinkan pengujian dilakukan secara berulang, konsisten, dan terdokumentasi dengan baik. Contoh potongan script pengujian menggunakan Katalon Studio ditunjukkan pada kode berikut.

```

WebUI.openBrowser("")
WebUI.navigateToUrl('https://dprd.bengkalis.go.id/webadmin/loginpage')

WebUI.setText(findTestObject('Page_Login/input_username'), username)
WebUI.setEncryptedText(findTestObject('Page_Login/input_password'), password)

WebUI.click(findTestObject('Page_Login/button_login'))

if (WebUI.verifyElementPresent(findTestObject('Page_Dashboard/dashboard'),
FailureHandling.OPTIONAL)) {
    WebUI.comment('Login berhasil')
} else {
    WebUI.comment('Login gagal')
}

WebUI.closeBrowser()
    
```

Skrip otomatis ini digunakan untuk mensimulasikan proses *login* pada sistem *website* DPRD Kabupaten Bengkalis. Pengujian dilakukan dengan menjalankan serangkaian perintah otomatis yang meliputi pembukaan *browser*, mengakses halaman *login*, memasukkan *username* dan *password*, serta menekan tombol *login* untuk memverifikasi proses autentikasi pengguna. Sistem kemudian merespon input yang diberikan dengan menampilkan halaman utama apabila login berhasil atau pesan kesalahan apabila data yang dimasukkan tidak *valid*.

5. Eksekusi Otomatisasi dan Pencatatan Hasil

Skrip pengujian yang telah dikembangkan kemudian dijalankan secara otomatis menggunakan Katalon Studio. Proses eksekusi dilakukan secara iteratif berdasarkan jumlah baris data pada file Excel, dimana setiap baris data diproses sebagai satu skenario pengujian. Dengan pendekatan ini, seluruh variasi data uji dapat dijalankan secara otomatis dalam satu siklus pengujian tanpa perlu intervensi manual.

a. Status keberhasilan atau kegagalan login

Setiap data uji yang dijalankan akan menghasilkan status yang menunjukkan apakah proses *login* berhasil atau tidak. Status ini secara otomatis tercatat dalam laporan pengujian Katalon Studio dan ditampilkan sebagai:

- a) *Passed*, apabila sistem berhasil memproses *login* dan pengguna diarahkan dari halaman *login* menuju halaman *dashboard admin*.
- b) *Failed*, apabila sistem menolak proses *login* sehingga pengguna tetap berada pada halaman *login*.

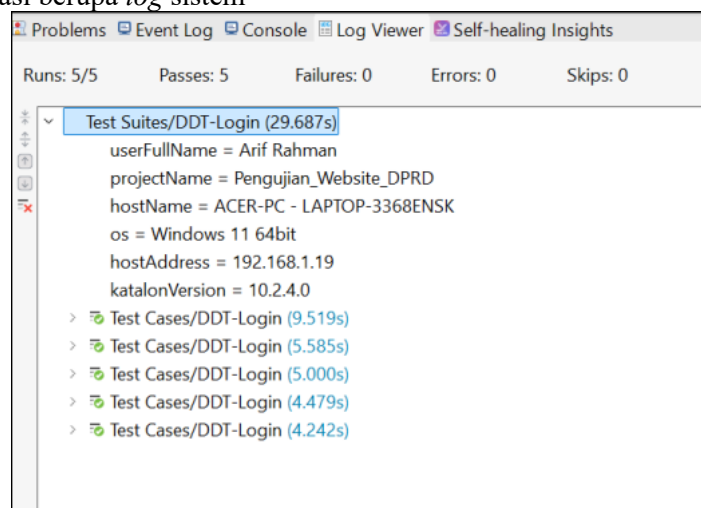
Karena sistem *login* tidak selalu menampilkan pesan kesalahan secara eksplisit, maka verifikasi keberhasilan *login* dalam penelitian ini ditentukan berdasarkan perubahan halaman setelah proses *login* dilakukan.

Selain itu, verifikasi hasil pengujian juga dilakukan dengan memanfaatkan fungsi validasi elemen pada Katalon Studio, yaitu dengan memeriksa keberadaan elemen tertentu pada halaman dashboard sebagai indikator keberhasilan login. Pendekatan ini digunakan untuk meningkatkan akurasi verifikasi hasil pengujian.

b. Waktu eksekusi pengujian

Selain mencatat status hasil pengujian, Katalon Studio juga mencatat waktu eksekusi untuk setiap skenario pengujian. Waktu eksekusi menunjukkan durasi yang diperlukan sistem untuk memproses permintaan *login*, mulai dari membuka halaman *login*, memasukkan *data username* dan *password*, menekan tombol *login*, hingga sistem memberikan respon. Informasi waktu eksekusi ini membantu peneliti dalam mengetahui efisiensi proses autentikasi yang dijalankan oleh sistem selama proses pengujian otomatis.

c. Bukti validasi berupa *log* sistem



Gambar 2 Validasi log sistem

Sebagai bukti validasi, Katalon Studio secara otomatis menghasilkan *log* pengujian setelah seluruh skenario dijalankan. *Log* ini berisi informasi mengenai urutan langkah pengujian, waktu eksekusi, serta status keberhasilan atau kegagalan dari setiap skenario yang dijalankan. Untuk menjaga keamanan *data*, informasi kredensial yang digunakan dalam proses pengujian tidak ditampilkan secara lengkap dalam laporan pengujian.

6. Evaluasi Hasil dan Dokumentasi

Setelah seluruh skenario pengujian dijalankan, tahap selanjutnya adalah melakukan evaluasi terhadap hasil pengujian yang diperoleh. Proses evaluasi dilakukan dengan membandingkan hasil keluaran sistem dengan hasil yang diharapkan pada setiap kombinasi data uji yang digunakan. Pada penelitian ini, evaluasi difokuskan pada pengujian form login admin sebagai contoh implementasi metode *Data-Driven Testing* pada sistem website DPRD Kabupaten Bengkalis.

Setiap hasil pengujian kemudian dicatat dalam bentuk tabel pengujian yang memuat *data input*, hasil yang diharapkan, hasil aktual sistem, serta status keberhasilan pengujian. Dokumentasi ini bertujuan untuk memastikan bahwa sistem telah memproses setiap variasi input sesuai dengan aturan validasi yang diterapkan. Hasil evaluasi dan dokumentasi pengujian ini selanjutnya menjadi dasar dalam penyusunan bagian Hasil dan Pembahasan, yang menjelaskan performa sistem dalam menangani berbagai variasi data pada proses *login admin*.

3. HASIL DAN PEMBAHASAN

Berdasarkan hasil eksekusi pengujian otomatis menggunakan Katalon Studio, dari total 11 test case yang dijalankan pada seluruh modul sistem, diperoleh hasil sebagai berikut:

- Test case berhasil (passed): 1 test case (9,1%)
- Test case gagal (failed): 10 test case (90,9%)

Hasil ini menunjukkan bahwa sebagian besar modul pada sistem masih memiliki kelemahan pada mekanisme validasi input. Waktu rata-rata eksekusi pengujian otomatis untuk satu siklus pengujian adalah sekitar 15–25 detik. Dibandingkan dengan pengujian manual, pendekatan otomatisasi ini tetap memberikan efisiensi dalam proses pengujian meskipun banyak ditemukan kesalahan pada sistem.

Berdasarkan proses evaluasi hasil pengujian yang telah dilakukan pada tahap sebelumnya, pengujian terhadap *form login admin* dilakukan menggunakan metode *Data-Driven Testing* dengan bantuan Katalon Studio. Pengujian ini bertujuan untuk memastikan bahwa mekanisme autentikasi dan validasi *input* pada sistem berjalan sesuai dengan skenario pengujian yang telah dirancang. Pengujian dilakukan melalui beberapa skenario, yaitu login dengan *username* dan *password* yang *valid*, login dengan *username* salah dan *password* benar, login dengan *username* benar dan *password* salah, login dengan *username* dan *password* yang keduanya salah, serta kondisi *username* dan *password* dikosongkan. Setiap skenario dijalankan secara otomatis dengan memanfaatkan variasi data uji yang telah disusun sebelumnya.

Hasil pengujian menunjukkan bahwa sistem mampu memberikan respons yang sesuai pada setiap skenario tersebut. Sistem menerima akses ketika data yang dimasukkan *valid* serta menolak akses ketika data yang dimasukkan tidak *valid* atau tidak lengkap. Dengan demikian, dapat disimpulkan bahwa mekanisme autentikasi dan validasi pada *form login* telah berjalan dengan baik dan sesuai dengan kebutuhan sistem. Meskipun penelitian ini menampilkan pengujian pada form login sebagai contoh implementasi metode *Data-Driven Testing*, proses pengujian juga dilakukan pada beberapa form lain yang terdapat pada sistem *admin website* DPRD Kabupaten Bengkalis. Hasil evaluasi secara keseluruhan menunjukkan bahwa masih terdapat beberapa form yang memerlukan perbaikan, khususnya pada mekanisme validasi input.

Beberapa form masih memungkinkan penyimpanan data meskipun terdapat field yang bersifat wajib tidak diisi atau terisi secara otomatis melalui nilai bawaan. Tingginya jumlah kegagalan pengujian menunjukkan bahwa sistem belum menerapkan validasi input secara optimal pada sebagian besar modul. Hal ini disebabkan oleh tidak adanya validasi terhadap field wajib (required field), serta adanya nilai default yang secara otomatis terisi tanpa interaksi pengguna.

Analisis lebih lanjut menunjukkan bahwa sebagian besar kesalahan terjadi akibat tidak adanya pengecekan input kosong (null validation) serta kurangnya pembatasan format data pada beberapa field penting. Kondisi ini berpotensi menyebabkan:

- tersimpannya data yang tidak lengkap,

- inkonsistensi data antar modul,
- menurunnya keakuratan informasi yang disajikan oleh sistem.

Apabila perbaikan tidak dilakukan, sistem berpotensi menghasilkan data yang tidak lengkap dan tidak konsisten sehingga dapat memengaruhi keakuratan informasi yang disajikan serta mengganggu proses pengelolaan data oleh admin.

Ringkasan hasil evaluasi pengujian pada sistem admin website DPRD Kabupaten Bengkalis ditunjukkan pada Tabel 4.

Tabel 4 Evaluasi Hasil Pengujian

Form yang Diuji	Temuan Utama	Rekomendasi Perbaikan	Dampak jika Tidak Diperbaiki
Login	Tidak ditemukan kesalahan validasi	Tidak diperlukan perbaikan	-
Informasi Publik	Sistem menyimpan data meskipun <i>field</i> keterangan dan file download kosong	Perlu perbaikan validasi pada <i>field</i> wajib	Data informasi publik tidak lengkap
Polling	Status tersimpan meskipun tidak dipilih karena nilai bawaan	Validasi status harus memastikan pilihan pengguna	Hasil polling tidak akurat
Agenda Kegiatan	Data tetap tersimpan meskipun beberapa <i>field</i> wajib kosong	Validasi wajib pada agenda, tanggal, jam, dan lokasi	Informasi jadwal kegiatan tidak sesuai
Pengumuman	Isi content kosong tetap disimpan	Validasi <i>field</i> isi content	Informasi pengumuman tidak lengkap
Kepegawaian	Banyak <i>field</i> wajib kosong tetap tersimpan	Validasi ketat pada seluruh <i>field</i> wajib	Data kepegawaian tidak akurat dan tidak lengkap
Berita	<i>Field</i> penting dan format gambar tidak divalidasi	Validasi <i>field</i> wajib dan format file	Kualitas informasi berita menurun
Berita Foto	<i>Field</i> anggota dewan dan fotografer tidak divalidasi	Penegasan <i>field</i> wajib	Data dokumentasi tidak jelas
Video	<i>Field</i> wajib dan format gambar tidak divalidasi	Validasi <i>field</i> dan format file	Informasi video tidak lengkap
Galeri	<i>Field</i> fotografer dan anggota dewan tidak divalidasi	Validasi <i>field</i> wajib	Data galeri tidak lengkap
Agenda Protokoler	Banyak <i>field</i> wajib kosong tetap tersimpan	Validasi seluruh <i>field</i> wajib	Informasi agenda tidak sesuai

Berdasarkan tingkat dampak yang ditimbulkan, prioritas perbaikan difokuskan pada modul kepegawaian, agenda kegiatan, dan informasi publik karena memiliki jumlah temuan kesalahan paling banyak. Perbaikan pada modul tersebut diharapkan dapat meningkatkan kualitas data serta keandalan sistem secara keseluruhan.

Penerapan Data-Driven Testing dalam penelitian ini terbukti mampu meningkatkan efisiensi pengujian dibandingkan metode manual. Dengan menggunakan satu skrip yang terhubung dengan berbagai data uji, proses pengujian dapat dilakukan secara berulang dan konsisten tanpa intervensi manual. Pendekatan ini juga meningkatkan cakupan pengujian serta meminimalkan kesalahan manusia dalam proses pengujian.

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, penerapan metode *Data-Driven Testing* pada sistem *admin website* DPRD Kabupaten Bengkalis dapat membantu proses pengujian perangkat lunak dilakukan secara lebih sistematis dan efisien. Integrasi data uji yang disimpan dalam *file Microsoft Excel* dengan skrip otomatis pada Katalon Studio memungkinkan berbagai kombinasi input diuji secara berulang tanpa perlu melakukan perubahan pada skrip pengujian. Hasil pengujian pada *form login* menunjukkan bahwa mekanisme autentikasi dan validasi *input* telah berjalan dengan baik, dimana sistem hanya memberikan akses ketika kredensial yang dimasukkan valid serta menolak akses ketika data tidak sesuai atau tidak lengkap. Hal ini menunjukkan bahwa fungsi autentikasi pada sistem telah berjalan sesuai dengan kebutuhan fungsional yang ditetapkan.

Namun demikian, hasil evaluasi terhadap beberapa form lain pada sistem *admin* menunjukkan masih terdapat kelemahan pada mekanisme validasi *input*. Beberapa form masih memungkinkan penyimpanan data meskipun terdapat *field* yang bersifat wajib tidak diisi atau terisi secara otomatis melalui nilai bawaan. Kondisi tersebut menunjukkan bahwa penerapan aturan validasi *input* pada sistem belum sepenuhnya optimal. Secara keseluruhan, penelitian ini menunjukkan bahwa penerapan *Data-Driven Testing* tidak hanya meningkatkan efisiensi proses pengujian, tetapi juga mampu membantu mengidentifikasi kelemahan pada mekanisme validasi sistem. Hasil penelitian ini diharapkan dapat menjadi dasar evaluasi bagi pengelola sistem dalam meningkatkan kualitas dan keandalan website admin DPRD Kabupaten Bengkalis.

REFERENCES

- [1] I. M. Jelantik, N. W. Utami, and I. Gst. A. P. Dwi, “Rancang Bangun Website Pemerintahan Desa Guwang sebagai Media Informasi Desa dan Promosi Destinasi Wisata,” *Jurnal Pengabdian Nasional (JPN) Indonesia*, vol. 5, no. 3, pp. 639–648, Sep. 2024, doi: 10.35870/jpni.v5i3.1021.
- [2] S. J. Krieg *et al.*, “Data-driven testing program improves detection of COVID-19 cases and reduces community transmission,” *NPJ Digit. Med.*, vol. 5, no. 1, pp. 1–5, Dec. 2022, doi: 10.1038/s41746-022-00562-4.
- [3] M. Anggi Puspita, M. Mardiana, R. A. Pradipta, and M. A. Muhammad, “User Acceptance Testing dan Performance Testing Pada Pengembangan Website Kanal Pengetahuan Dikti,” *JIEET: (Journal Information Engineering and Educational Technology)*, vol. 08, no. 02, pp. 96–101, 2024, [Online]. Available: <https://kanalpengetahuansatu.dikti.go.id>. [2]
- [4] A. Arifandi, R. N. Z. Simamora, G. A. Janitra, M. A. Yaqin, and M. M. Huda, “Survei Teknik-Teknik Pengujian Software Menggunakan Metode Systematic Literature Review,” *ILKOMNIKA: Journal of Computer Science and Applied Informatics E*, vol. 4, no. 3, pp. 297–315, Dec. 2022, doi: 10.28926/ilkomnika.v4i3.436.
- [5] I. R. Dhaifullah, M. M. H, A. A. Salsabila, and M. A. Yakin, “Survei Teknik Pengujian Software,” *JACIS: Journal Automation Computer Information System*, vol. 2, no. 1, pp. 31–38, May 2022.
- [6] K. Wijaya and T. W. Harjanti, “Quality Assurance Pada Sistem ERP Equip Modul Purchase dan Inventory di PT. Hashmicro Solusi Indonesia,” *Jurnal Maklumatika*, vol. 8, no. 1, pp. 58–68, Jul. 2021.
- [7] V. A. Amel, A. W. R. Emanuel, and S. P. Adithama, “Pengujian Website ACC.CO.ID Revamp Menggunakan Metode Black Box Testing,” *Jurnal Informatika Atma Jogja*, vol. 2, no. 1, pp. 39–46, May 2021.

- [8] M. P. A. Ginting and A. S. Lubis, “Pengujian Aplikasi Berbasis Web Data Ska Menggunakan Metode Black Box Testing,” *COSMIC JURNAL TEKNIK*, vol. 2, no. 1, pp. 41–48, Feb. 2024, doi: 10.55537/cosmic.
- [9] S. D. Pratama and M. N. Dadaprawira, “Pengujian Black Box Testing Pada Aplikasi Edu Digital Berbasis Website Menggunakan Metode Equivalence Dan Boundary Value,” *Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD*, vol. 6, no. 2, pp. 560–569, Jul. 2023, [Online]. Available: <https://ojs.trigunadharma.ac.id/index.php/jsk/index>
- [10] M. Mintarsih, “Pengujian Black Box Dengan Teknik Transition Pada Sistem Informasi Perpustakaan Berbasis Web Dengan Metode Waterfall Pada SMC Foundation,” *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 5, no. 1, pp. 33–35, Feb. 2023, doi: 10.47233/jteksis.v5i1.727.
- [11] D. K. P. Wahyudi, “Black Box Testing Aplikasi Point Of Sales Post,” *KURAWAL Jurnal Teknologi, Informasi dan Industri*, vol. 4, no. 1, pp. 1–16, Mar. 2021, [Online]. Available: <https://jurnal.machung.ac.id/index.php/kurawal>
- [12] M. Rafli, I. Fitri, and A. Andrianingsih, “Pengujian Kinerja Load Balancing Web Server Menggunakan Nginx Reverse Proxy Berbasis OS Centos 7,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 9, no. 3, pp. 1824–1840, Sep. 2022, [Online]. Available: <http://jurnal.mdp.ac.id>
- [13] P. K. koppanati, “Implementing Data-Driven Testing in Java and JavaScript Using JSON and XML,” *International Journal For Multidisciplinary Research*, vol. 3, pp. 1–6, Nov. 2021, [Online]. Available: www.ijfmr.com
- [14] M. T. Banjarnahor and L. S. Istiyowati, “Otomatisasi Pengujian Perangkat Lunak menggunakan Metode Data Driven untuk meningkatkan efisiensi waktu pengujian,” 2022.
- [15] U. Sa’adah *et al.*, “Automatic Testing Framework Based On Serenity And Jenkins Automated Build,” *JUTI: Jurnal Ilmiah Teknologi Informasi*, vol. 19, pp. 102–110, Jul. 2021, doi: 10.12962/j24068535.v19i2.a1017.
- [16] M. Neelapu, “Meningkatkan Cakupan Pengujian melalui Pendekatan Otomatisasi Berbasis Data,” *International Journal of Multidisciplinary Research and Growth Evaluation*, vol. 05, pp. 1684–1691, Jan. 2024.
- [17] A. B. Bujok, “Development of a Software Testing Best Practice Framework for Medical Device Software,” Doctoral Thesis, Dundalk Institute of Technology, Dundalk, 2021. Accessed: Mar. 05, 2026. [Online]. Available: <https://eprints.dkit.ie/id/eprint/743>