

ANALISIS PENERAPAN *BLACK BOX TESTING* BERBASIS *DECISION TABLE* TERHADAP FUNGSIONALITAS WEBSITE FISHCO

Fadilla Julianifa Putri ¹, Dwi Febriyanti ², Dewi Wulansari ³,
Keisha Ramadhani ⁴, Aditya Wicaksono ⁵, Gema Parasti Mindara ⁶
^{1,2,3,4,5} Teknologi Rekayasa Perangkat Lunak, Sekolah Vokasi, IPB University
⁶ Teknologi Rekayasa Komputer, Sekolah Vokasi, IPB University
Jalan Kumbang No.14, Kota Bogor, Indonesia
adityawicaksono@apps.ipb.ac.id

ABSTRAK

Pengujian perangkat lunak merupakan tahapan penting dalam pengembangan sistem untuk menjamin kualitas dan keandalan sebelum digunakan secara luas. Website Fishco dikembangkan sebagai sistem informasi berbasis web yang memiliki aturan bisnis dan validasi *input* yang kompleks, sehingga membutuhkan metode pengujian yang tepat dan terstruktur. Permasalahan yang dihadapi adalah potensi kesalahan logika pada proses validasi *input* yang dapat memengaruhi fungsionalitas sistem apabila tidak diuji secara menyeluruh. Oleh karena itu, penelitian ini bertujuan untuk menganalisis efektivitas penerapan *Black Box Testing* berbasis *Decision Table Testing* dalam mengevaluasi fungsionalitas utama Website Fishco. Metode penelitian yang digunakan meliputi analisis kebutuhan dan kajian literatur, pengumpulan data, perancangan tabel keputusan, pelaksanaan pengujian, serta evaluasi hasil pengujian. Hasil penelitian menunjukkan bahwa seluruh skenario pengujian yang dirancang berdasarkan tabel keputusan dapat dijalankan dengan baik dan menghasilkan keluaran yang sesuai dengan aturan bisnis yang telah ditetapkan. Temuan ini membuktikan bahwa *Decision Table Testing* efektif dalam mendeteksi kesalahan logika validasi *input* serta memastikan fungsionalitas sistem berjalan sesuai kebutuhan pengguna, sehingga berkontribusi dalam meningkatkan kualitas dan keandalan Website Fishco.

Kata kunci : *black box testing, decision table testing, fishco, pengujian perangkat lunak, website*

1. PENDAHULUAN

Kualitas dan fungsionalitas aplikasi berbasis web merupakan faktor krusial yang menentukan keberhasilan adopsi dan kepuasan pengguna dalam konteks teknologi informasi modern [1]. Dalam pemenuhan kebutuhan informasi spesifik terkait dunia akuatik, dikembangkanlah Website Fishco yang dirancang untuk memfasilitasi para pemilik ikan hias agar lebih memahami dan mengelola ikan yang mereka miliki. Mengingat peran Fishco sebagai sumber rujukan dan alat bantu navigasi bagi pengguna, keakuratan fungsionalitas dan keandalan sistem (*system quality*) menjadi prasyarat mutlak untuk menjamin pengalaman pengguna yang positif dan tingkat adopsi (*usage continuance*) yang tinggi. Sejalan dengan hal tersebut, kualitas sistem terbukti berperan penting dalam meningkatkan kepuasan pengguna sekaligus mempertahankan penggunaan layanan digital [2]. Dengan demikian, pengujian sistem yang komprehensif menjadi langkah penting untuk memastikan keandalan sistem, ketepatan pengolahan data, serta tercapainya pengalaman fungsional yang optimal bagi pengguna [3].

Selain memastikan kualitas dan keandalan fitur inti pada Website Fishco, keberhasilan sebuah aplikasi web juga sangat dipengaruhi oleh kualitas sistem, kualitas layanan, dan kualitas informasi yang disediakan. Penelitian sebelumnya menunjukkan bahwa aspek-aspek tersebut memiliki kontribusi signifikan terhadap tingkat kepuasan pengguna serta keberlanjutan penggunaan layanan digital [4]. Dengan demikian, pemenuhan validasi fungsional yang tepat

serta pengujian sistematis menjadi komponen yang tidak dapat dipisahkan dalam pengembangan aplikasi berbasis web [5]. Dalam Website Fishco, pengujian sistem tidak hanya bertujuan untuk memastikan setiap fitur berjalan sesuai dengan logika yang telah dirancang, tetapi juga untuk meminimalkan potensi kesalahan fungsional yang dapat berdampak pada akurasi informasi dan kepercayaan pengguna. Hal ini menjadi sangat penting terutama pada fitur-fitur kritis yang melibatkan aturan validasi yang ketat dan saling berkaitan.

Meskipun demikian, saat ini belum adanya pengujian secara sistematis yang mampu memetakan seluruh kombinasi kondisi dan aksi pada validasi fungsional Website Fishco, khususnya pada fitur-fitur yang memiliki aturan domain ikan hias yang kompleks. Kompleksitas aturan tersebut berpotensi menimbulkan kesalahan logika validasi apabila tidak diuji menggunakan pendekatan yang terstruktur. Oleh karena itu, diperlukan suatu metode pengujian yang mampu merepresentasikan berbagai kombinasi kondisi dan aksi secara sistematis guna memastikan seluruh skenario *input* dan *output* dapat tervalidasi dengan baik. Berdasarkan uraian tersebut, rumusan masalah dalam penelitian ini adalah bagaimana memastikan keandalan validasi fungsional Website Fishco yang memiliki aturan domain kompleks melalui penerapan metode pengujian yang sistematis.

Dalam hal ini, *Decision Table Testing* dipandang sebagai teknik yang paling efektif untuk menangkap seluruh skenario *input-output* yang mungkin terjadi dalam sistem dengan aturan logika yang kompleks [3],

[6]. Beberapa penelitian terdahulu telah membuktikan efektivitas metode ini. Pengujian pada Website Pusat Studi Bencana IPB menunjukkan kinerja yang memuaskan dalam memenuhi kebutuhan fungsional maupun non-fungsional [3]. Selain itu, pengujian pada Website Infotech menggunakan teknik *Black-Box Testing* dengan metode *Decision Table Testing* menyimpulkan bahwa fitur form login dan form submit file telah beroperasi sesuai spesifikasi tanpa ditemukan adanya *bug* [5].

Berdasarkan kondisi tersebut, penelitian ini bertujuan untuk menerapkan metode *Black Box Testing* berbasis *Decision Table Testing* dalam mengevaluasi fungsionalitas utama Website Fishco. Secara khusus, penelitian ini bertujuan untuk mengidentifikasi kesesuaian antara kebutuhan fungsional dan implementasi sistem, serta menilai efektivitas *Decision Table Testing* dalam memvalidasi fitur-fitur dengan aturan domain yang ketat dan kompleks.

Sebagai upaya penyelesaian masalah, penelitian ini dilakukan melalui beberapa tahapan. Tahapan tersebut dimulai dengan mengidentifikasi kebutuhan fungsional pada Website Fishco. Selanjutnya, dilakukan penyusunan tabel keputusan untuk memetakan kombinasi kondisi dan aksi pada sistem. Pengujian dilakukan berdasarkan skenario yang dihasilkan, kemudian hasil pengujian dianalisis untuk menentukan tingkat kesesuaian fungsi sistem. Pendekatan ini diharapkan mampu memastikan keandalan fungsional Website Fishco sekaligus memberikan kontribusi praktis sebagai referensi pengujian bagi website lain dengan karakteristik yang serupa.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Pemanfaatan *Black Box Testing* berbasis *Decision Table Testing* sudah banyak dilakukan oleh penelitian sebelumnya untuk melakukan pengujian fungsionalitas sistem berbasis web maupun aplikasi. Pada tahun 2024 dilakukan penerapan *Decision Table Testing* pada Sistem Aplikasi *Mobile Sharing Story App* dan menunjukkan bahwa teknik ini efektif untuk memastikan fungsi utama seperti registrasi dan login berjalan sesuai dengan aturan skenario yang dibuat dalam tabel keputusan [7]. Di tahun yang sama, telah dilakukan penerapan *Decision Table Testing* pada website Pusat Studi Bencana IPB untuk mengidentifikasi potensi kesalahan pada *input-output* fungsi sehingga pengujian menjadi lebih sistematis dan komprehensif [3]. Selain itu, di tahun 2022 juga menunjukkan efektivitas *Decision Table Testing* dalam pengujian fitur login dan unggah file pada website Infotech, di mana metode ini dinilai berhasil mengevaluasi fungsionalitas utama sesuai dengan aturan bisnis yang ditetapkan [5].

Berdasarkan hasil penelitian terdahulu yang dijabarkan, *Decision Table Testing* terbukti unggul dalam menangani pengujian fungsional dengan aturan

dan kombinasi *input* yang kompleks. Oleh karena itu, metode ini menjadi metode yang tepat untuk pengujian Website Fishco karena mampu menguji fungsionalitas berbasis aturan secara sistematis dan menyeluruh.

2.2. Software Testing

Pengujian perangkat lunak (*Software Testing*) adalah proses krusial dalam pengembangan perangkat lunak yang bertujuan untuk memastikan kualitas, kinerja dan keamanan dari produk yang dikembangkan [8]. Produk ini dapat berupa sistem berbasis web, *mobile apps*, atau perangkat lunak lainnya. Menurut [9], *Software Testing* didefinisikan sebagai suatu proses menjalankan program dengan maksud menemukan kesalahan. Tujuan ini dicapai melalui berbagai pendekatan pengujian yang berbeda, seperti *White Box Testing* dan *Black Box Testing*.

2.3. Black Box Testing

Black Box testing merupakan salah satu pendekatan pengujian perangkat lunak. Pendekatan ini dipakai untuk menguji sebuah perangkat lunak tanpa harus memperhatikan detail *software* [10]. Pengujian ini mencakup analisis fungsi sistem dan pemetaan alur kerja pengguna untuk menentukan area yang akan diuji secara fungsional [11]. Saat ini *Black Box Testing* menjadi pendekatan yang paling umum digunakan. Metode ini memastikan proses pengujian berjalan dengan akurat dan mendukung keandalan kinerja sistem secara menyeluruh [12]. Dalam *Black Box Testing*, terdapat berbagai teknik pengujian yang dapat digunakan, di antaranya *Equivalence Partitioning Testing*, *Boundary Value Analysis Testing*, *Comparison Testing*, *Sample Testing*, *Robustness Testing*, serta teknik pengujian lainnya [13].

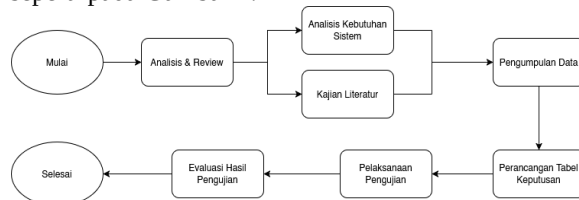
2.4. Decision Table Testing

Decision Table Testing (DTT) merupakan metode pada *Black Box Testing* yang digunakan untuk menguji perangkat lunak berdasarkan kombinasi kondisi dan aksi dari berbagai *input* dan *output* [6]. *Decision Table Testing* dimanfaatkan dalam pengujian untuk menentukan data masukan yang valid agar sistem dapat memproses dan melakukan perhitungan sesuai dengan ketentuan yang telah ditetapkan [14]. Metode ini membantu merangkum seluruh kemungkinan kondisi ke dalam sebuah tabel sehingga proses pengujian dapat dilakukan secara sistematis dan menyeluruh [7].

3. METODE PENELITIAN

Penelitian ini dilakukan melalui beberapa langkah penting yang bertujuan untuk menguji dan memastikan Website Fishco berfungsi dengan baik. Penelitian ini dimulai dengan menganalisis kebutuhan website untuk menentukan pendekatan yang paling efektif dalam menguji fungsionalitas Website Fishco. Pendekatan yang digunakan pada penelitian ini adalah pendekatan *Black Box Testing* dengan metode *Decision Table Testing* dengan fokus berdasarkan

spesifikasi dan kebutuhan pengguna akhir [5]. Penelitian kemudian dilanjutkan dengan perancangan tabel keputusan untuk memetakan kondisi dan aksi yang diuji. Selanjutnya, dilakukan pengujian berdasarkan tabel keputusan yang telah dirancang dan diakhiri dengan evaluasi hasil pengujian guna memastikan semua fungsi berjalan sesuai aturan bisnis yang berlaku. Adapun tahapan metodologi penelitian seperti pada Gambar 1.



Gambar 1. Tahap penelitian

3.1. Analisis dan Review

Tahap pertama adalah memahami kebutuhan sistem serta menentukan pendekatan dan metode pengujian yang sesuai dengan karakteristik Website Fishco. Kegiatan pada tahap ini meliputi analisis kebutuhan sistem dan kajian literatur. Analisis kebutuhan sistem bertujuan untuk mengidentifikasi fungsi utama, alur kerja sistem, serta kebutuhan pengguna. Langkah ini mencakup penentuan fitur yang akan diuji, data masukan (*input*) dan keluaran (*output*) yang diharapkan. Berdasarkan hasil analisis awal, pengujian dilakukan menggunakan pendekatan *Black Box Testing* dengan metode *Decision Table Testing* terhadap fungsi utama Website Fishco. Pengujian *black box* dilakukan untuk menganalisis fungsi sistem dan pemetaan alur kerja pengguna untuk menentukan area yang akan diuji secara fungsional [11].

Analisis dilakukan dengan meninjau dan mengidentifikasi proses bisnis, alur kerja pengguna, serta logika validasi *input* yang diterapkan pada sistem tanpa mengakses *source code*. Hasil dari tahap ini berupa daftar fitur yang akan diuji serta kondisi dan *output* yang diharapkan, sebagai dasar dalam penyusunan skenario pengujian. Selanjutnya dilakukan kajian literatur dari berbagai sumber seperti *e-book*, jurnal ilmiah, dan penelitian terdahulu yang berkaitan dengan pengujian perangkat lunak. Tujuan kajian literatur ini adalah untuk memahami konsep dasar serta penerapan berbagai pengujian perangkat lunak seperti *Black Box Testing*, *White Box Testing*, dan berbagai pendekatan turunannya.

3.2. Pengumpulan Data

Tahap pengumpulan data dilakukan dengan mengamati respons sistem saat menerima berbagai kombinasi *input* pada form yang sedang diuji. Data yang dikumpulkan meliputi kondisi *input*, hasil *output*, pesan *error* yang muncul, serta respons sistem terhadap tindakan pengguna. Langkah ini menekankan pentingnya pengumpulan data observasional dalam mengidentifikasi kesesuaian antara *input* dan *output* sistem selama pengujian fungsional menggunakan

Decision Table Testing [5]. Data yang telah dikumpulkan kemudian digunakan sebagai dasar dalam penyusunan *Decision Table Testing*.

3.3. Perancangan Tabel Keputusan

Pada tahap ini, dilakukan pemetaan hubungan antara kondisi (*input*) dan aksi (*output*) untuk menyusun *Decision Table Testing*. Setiap kombinasi kondisi diuji untuk menentukan respons yang seharusnya dihasilkan oleh sistem, baik dalam bentuk pesan *error*, penolakan *input*, maupun keberhasilan penyimpanan data. *Decision Table Testing* disusun dengan sistematis agar seluruh kemungkinan logika sistem dapat terwakili. Pendekatan eksplisit digunakan pada form Login dan Artikel karena kondisinya sederhana, sedangkan pendekatan implisit diterapkan pada form User, Penyakit Ikan, dan Produk yang memiliki banyak *field*. Pendekatan implisit bertujuan untuk menjaga efisiensi pengujian tanpa mengurangi ketelitian hasil. Penyusunan tabel keputusan membantu memetakan logika sistem serta mengidentifikasi potensi kesalahan pada proses validasi *input* secara sistematis [7].

3.4. Pelaksanaan Pengujian

Proses pengujian dilakukan dengan menjalankan skenario pengujian berdasarkan *Decision Table Testing* pada setiap fitur Website Fishco yang akan diuji. Penguji memasukkan berbagai kombinasi *input* sesuai dengan kondisi yang tercantum dalam tabel, kemudian mencatat hasil yang muncul di sistem. Tahap ini memverifikasi apakah *output* yang dihasilkan sistem sesuai dengan aksi yang diharapkan dalam *Decision Table Testing*. Pengujian diulang hingga semua kondisi dan aksi dalam *Decision Table Testing* telah terverifikasi dengan benar [15].

3.5. Evaluasi Hasil Pengujian

Tahap terakhir adalah melakukan evaluasi terhadap hasil pengujian untuk menilai keakuratan sistem dalam menanggapi berbagai kondisi *input*. Hasil pengujian dibandingkan dengan skenario dalam *Decision Table Testing* untuk memastikan kesesuaian antara respons aktual pada sistem dengan respons yang diharapkan. Evaluasi hasil pengujian berperan penting dalam memastikan keandalan logika sistem dan mendeteksi potensi kesalahan yang luput dalam pengujian manual [16]. Hasil dari tahap ini adalah mengetahui kesesuaian sistem dengan kebutuhan fungsional pengguna dan efektifitas metode *Decision Table Testing* dalam mendeteksi kesalahan logika validasi. Temuan ini digunakan sebagai dasar dalam penarikan kesimpulan dan penyusunan rekomendasi untuk pengembangan serta pengujian lanjutan pada Website Fishco.

4. HASIL DAN PEMBAHASAN

Pengujian dilakukan pada Website Fishco, yaitu sebuah sistem informasi berbasis website yang dikembangkan untuk membantu para pemilik ikan hias

dalam merawat ikan yang mereka miliki. Website ini bertujuan untuk memberikan kemudahan dalam mendeteksi penyakit ikan, mengatur ekosistem akuarium, dan memberikan tips perawatan terbaik untuk berbagai jenis ikan. Melalui penerapan metode *Decision Table Testing*, pengujian difokuskan pada validasi form untuk memastikan setiap *required field* berfungsi sebagaimana mestinya. Selain itu, pengujian juga bertujuan untuk memastikan bahwa sistem mampu menampilkan *error message* yang sesuai ketika pengguna memasukkan data yang tidak valid atau tidak lengkap.

4.1. Pelaksanaan Pengujian

Pengujian dilakukan dengan pendekatan *Black Box Testing* menggunakan metode *Decision Table Testing*. Dalam penerapannya, terdapat dua pendekatan yang dapat dilakukan, yaitu eksplisit dan implisit. Pada pendekatan eksplisit, seluruh kombinasi kondisi dituliskan secara lengkap dalam tabel keputusan dengan mengacu pada rumus 2^n , di mana n adalah jumlah kondisi. Pendekatan ini memastikan bahwa setiap kemungkinan *input* diuji tanpa terlewat, meskipun jumlah kombinasinya bisa sangat banyak. Sementara itu, pada pendekatan implisit, kombinasi kondisi yang digunakan adalah yang relevan atau memiliki potensi menghasilkan hasil berbeda saat diuji. Pendekatan ini lebih efisien karena mampu mengurangi jumlah *test case* tanpa mengurangi cakupan logika yang diuji. Kedua pendekatan ini membantu memastikan bahwa sistem telah divalidasi secara menyeluruh, baik dari segi kelengkapan maupun efisiensi pengujian [17].

Proses pengujian dilakukan dengan menyiapkan *test case* yang dihasilkan dari tabel keputusan. Setiap aturan dalam tabel keputusan merepresentasikan satu skenario pengujian dengan kondisi dan aksi tertentu. Untuk form Login, pengujian dilakukan dengan pendekatan eksplisit, di mana seluruh kombinasi kondisi dari setiap *field* diuji berdasarkan jumlah variabel yang ada. Tahapan dimulai dengan mengidentifikasi setiap *field* yang bersifat wajib diisi (*required field*), yaitu *Email* dan *Password*. Selanjutnya, ditentukan aksi berupa *output* yang diharapkan berdasarkan setiap kondisi *input* yang dimasukkan oleh pengguna. Tabel 1 menyajikan hasil pengujian form Login menggunakan pendekatan eksplisit yang mencakup seluruh kombinasi kondisi dari *field Email* dan *Password*.

Tabel 1. DTT pengujian form login

	1	2	3	4
Conditions				
c1 = <i>Email</i>	N	Y	N	Y
c2 = <i>Password</i>	N	N	Y	Y
Actions				
a1 = Gagal	X	X	X	
a2 = Berhasil				X

Form User memiliki enam *field* yang bersifat *required* yaitu, Nama, *Email*, *Password*, Konfirmasi *Password*, No. Hp, dan Alamat. Seluruh *field* tersebut wajib diisi oleh pengguna untuk memastikan proses pendaftaran akun dapat berjalan dengan baik dan menghasilkan *output* yang sesuai. Apabila salah satu *field* tidak diisi atau tidak memenuhi aturan validasi, maka sistem harus menampilkan pesan kesalahan dan mencegah data untuk disimpan. Dengan banyaknya *field* yang terdapat pada form User, maka pengujian dilakukan melalui pendekatan implisit dengan memfokuskan pada kondisi-kondisi kritis yang paling berpotensi menyebabkan kegagalan sistem. Tabel 2 menyajikan hasil pengujian form User menggunakan pendekatan implisit dengan beberapa kombinasi.

Tabel 2. DTT pengujian form user

	1	2	3	4	5	6	7	8	9
Conditions									
c1 = Nama	N	N	Y	Y	Y	Y	Y	Y	Y
c2 = <i>Email</i>	N	Y	N	Y	Y	Y	Y	Y	Y
c3 = <i>Password</i>	N	Y	Y	N	Y	Y	Y	N	Y
c4 = Konfirmasi <i>Password</i>	N	Y	Y	Y	N	Y	Y	N	Y
c5 = No. Hp	N	Y	Y	Y	Y	N	Y	Y	Y
c6 = Alamat	N	Y	Y	Y	Y	Y	N	Y	Y
Actions									
a1 = Gagal	X	X	X	X	X	X	X	X	
a2 = Berhasil									X

Pada form Penyakit Ikan dilakukan pengujian menggunakan pendekatan implisit, di mana tidak semua kombinasi kondisi dari setiap *field* diuji secara penuh, melainkan hanya kombinasi yang relevan dan berpotensi menimbulkan hasil berbeda pada sistem. Pendekatan ini dipilih karena jumlah *field* yang banyak akan menghasilkan kombinasi kondisi yang sangat besar apabila diuji secara eksplisit. Tahapan pengujian dimulai dengan mengidentifikasi *required field* yang ada, yaitu Nama, Tipe Penyakit, Penyebab, Bagian yang terpengaruh, Deskripsi, Gejala, Pencegahan, dan *Note*. Sedangkan untuk *field* Ikan yang terjangkau dan Rekomendasi Produk tidak wajib diisi. Selanjutnya, ditentukan tindakan berupa *output* yang diharapkan berdasarkan kondisi *input* yang diberikan oleh pengguna, apakah sistem akan menampilkan pesan gagal atau berhasil menyimpan data. Dengan pendekatan implisit ini, pengujian difokuskan pada kombinasi kondisi yang paling mewakili kemungkinan kesalahan *input*, sehingga dapat mengoptimalkan proses validasi tanpa harus menguji seluruh kemungkinan yang ada. Tabel 3 menyajikan hasil pengujian form Penyakit Ikan menggunakan pendekatan implisit berdasarkan kombinasi kondisi dari setiap *field* yang relevan.

Tabel 3. DTT pengujian form penyakit ikan

	1	2	3	4	5	6	7	8	9	10
Conditions										
c1 = Nama	N	N	Y	Y	Y	Y	Y	Y	Y	Y
c2 = Tipe Penyakit	N	Y	N	Y	Y	Y	Y	Y	Y	Y
c3 = Penyebab	N	Y	Y	N	Y	Y	Y	Y	Y	Y
c4 = Bagian yang terpengaruh	N	Y	Y	Y	N	Y	Y	Y	Y	Y
c5 = Ikan yang terjangkit	N	Y	Y	Y	Y	Y	Y	Y	Y	N
c6 = Deskripsi	N	Y	Y	Y	Y	N	Y	Y	Y	Y
c7 = Gejala	N	Y	Y	Y	Y	Y	N	Y	Y	Y
c8 = Pencegahan	N	Y	Y	Y	Y	Y	Y	N	Y	Y
c9 = Note	N	Y	Y	Y	Y	Y	Y	Y	N	Y
c10 = Rekomendasi Produk	N	Y	Y	Y	Y	Y	Y	Y	Y	N
Actions										
a1 = Gagal	X	X	X	X	X	X	X	X	X	
a2 = Berhasil										X

Selanjutnya, pengujian pada form Produk menggunakan pendekatan serupa yaitu implisit dengan enam *required field*. Enam *required field* tersebut meliputi Nama, Kategori, Harga, Link Produk, Deskripsi Produk, dan *Upload Thumbnail*. Pendekatan implisit ini diterapkan untuk mengoptimalkan proses, mengingat tujuan utamanya adalah memvalidasi integritas pada *field* wajib. Dalam kasus ini, pengujian difokuskan pada delapan *rule* mencakup skenario keberhasilan (semua terisi), skenario kegagalan total (semua kosong), dan enam skenario kegagalan tunggal (dimana hanya satu *field* yang dikosongkan). Tabel 4 menyajikan hasil pengujian form Produk berdasarkan kombinasi *input* yang relevan.

Tabel 4. DTT pengujian form produk

	1	2	3	4	5	6	7	8
Conditions								
c1 = Nama	N	N	Y	Y	Y	Y	Y	Y
c2 = Kategori	N	Y	N	Y	Y	Y	Y	Y
c3 = Harga	N	Y	Y	N	Y	Y	Y	Y
c4 = Link Produk	N	Y	Y	Y	N	Y	Y	Y
c5 = Deskripsi Produk	N	Y	Y	Y	Y	N	Y	Y
c6 = Upload Thumbnail	N	Y	Y	Y	Y	Y	N	Y
Actions								
a1 = Gagal	X	X	X	X	X	X	X	
a2 = Berhasil								X

Pengujian dilakukan juga pada form Artikel dengan tiga *field* yang bersifat *required*, yaitu *Thumbnail*, Judul, dan Badan Artikel. Ketiga *field* ini wajib diisi oleh pengguna agar proses penyimpanan artikel dapat dilakukan oleh sistem. Apabila salah satu

field kosong, maka sistem harus menampilkan pesan kesalahan dan mencegah data untuk disimpan. Melihat jumlah *field* yang relatif sedikit maka pengujian dilakukan dengan pendekatan eksplisit. Tabel 5 menyajikan hasil pengujian form Artikel menggunakan pendekatan eksplisit yang mencakup seluruh kombinasi kondisi dari *field Thumbnail*, Judul, dan Badan Artikel.

Tabel 5. DTT pengujian form artikel

	1	2	3	4	5	6	7	8
Conditions								
c1 = Thumbnail	N	Y	N	N	Y	Y	N	Y
c2 = Judul	N	N	Y	N	Y	N	Y	Y
c3 = Badan Artikel	N	N	N	Y	N	Y	Y	Y
Actions								
a1 = Gagal	X	X	X	X	X	X	X	
a2 = Berhasil								X

4.2. Interpretasi Hasil

Berdasarkan hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa sistem pada Website Fishco telah memberikan respons yang sesuai terhadap setiap kombinasi kondisi *input* yang diuji. Pada form Login, hasil pengujian menunjukkan bahwa sistem mampu menolak *input* kosong serta hanya menerima *input* yang benar dan lengkap. Hal ini membuktikan bahwa validasi pada form Login telah berjalan dengan baik sesuai dengan logika tabel keputusan. Tidak ditemukan *error* pada proses autentikasi, sehingga sistem dapat dikatakan berhasil menangani seluruh skenario pengujian eksplisit secara akurat.

Pada form User, seluruh *field* seperti Nama, Email, Password, Konfirmasi Password, No. Hp, dan Alamat bersifat wajib diisi. Oleh karena itu, validasi harus memastikan tidak adanya *input* kosong maupun ketidaksesuaian format, seperti ketidaksesuaian antara Password dan Konfirmasi Password. Hasil pengujian melalui pendekatan implisit menunjukkan bahwa sistem selalu memberikan aksi “Gagal” pada seluruh skenario ketika salah satu *field* dikosongkan atau ketika terdapat kombinasi kondisi yang tidak memenuhi aturan validasi. Hanya pada *rule* ke-9, ketika seluruh *field* diisi dengan benar, sistem memberikan aksi “Berhasil.” Hal ini membuktikan bahwa validasi pada form User telah berjalan konsisten, dan tabel keputusan berhasil menangkap aturan kritis yang paling mempengaruhi keberhasilan pendaftaran akun.

Sementara itu, pada form Penyakit Ikan, hasil pengujian menunjukkan bahwa sistem mampu menolak pengisian form yang belum lengkap dan menampilkan *error message* “Please fill in this field.” Pesan tersebut muncul ketika salah satu *required field*, yaitu Nama, Tipe Penyakit, Penyebab, Bagian yang terpengaruh, Deskripsi, Gejala, Pencegahan, atau Note tidak diisi. Hal ini menandakan bahwa mekanisme validasi *input* telah berjalan sesuai dengan logika tabel keputusan yang dirancang. Pada kombinasi kondisi

ketika seluruh *required field* telah diisi dengan benar, sistem memberikan tindakan “Berhasil”, meskipun *field* opsional seperti Ikan yang Terjangkit dan Rekomendasi Produk tidak diisi, sehingga data tetap dapat disimpan ke dalam sistem. Hasil ini menunjukkan bahwa sistem mampu membedakan dengan tepat antara *field* wajib dan *field* opsional sesuai dengan aturan validasi yang ditetapkan. Pendekatan implisit terbukti efisien karena mampu merepresentasikan logika utama validasi *input* tanpa harus menguji seluruh kemungkinan kombinasi kondisi. Dengan memfokuskan pengujian pada kombinasi yang paling relevan dan berpotensi memengaruhi hasil validasi, metode ini tetap dapat memastikan keandalan sistem dalam mendeteksi kesalahan *input* secara cepat dan akurat. Selain itu, hasil pengujian menunjukkan bahwa penerapan pendekatan implisit sangat sesuai untuk pengujian form dengan jumlah *field* yang banyak seperti pada form Penyakit Ikan, karena mampu menjaga efisiensi waktu dan penggunaan sumber daya pengujian tanpa mengurangi ketelitian hasil.

Selanjutnya, pada form Produk yang memiliki enam *required field* yaitu Nama, Kategori, Harga, Link Produk, Deskripsi Produk, dan *Upload Thumbnail*, pendekatan implisit juga diterapkan untuk memastikan tiap aturan utama validasi dapat diuji secara efisien. Hasil pengujian menunjukkan bahwa sistem menolak seluruh kombinasi *input* ketika salah satu *field* dikosongkan, dan hanya memberikan aksi “Berhasil” pada *rule* terakhir ketika seluruh *field* terisi lengkap dan sesuai aturan. Konsistensi hasil ini menegaskan bahwa mekanisme validasi pada form Produk telah berfungsi sebagaimana mestinya.

Pada form Artikel, hasil pengujian menunjukkan bahwa sistem selalu memberikan aksi “Gagal” pada setiap *rule* ketika salah satu atau lebih *field* dikosongkan, yang menandakan bahwa validasi berjalan sesuai aturan dan mampu mendeteksi *input* yang tidak lengkap. Hanya pada kombinasi terakhir, ketika ketiga *field* terisi dengan benar, sistem memberikan aksi “Berhasil” dan mengizinkan data disimpan. Hal ini membuktikan bahwa logika validasi pada form Artikel telah berfungsi secara konsisten dan responsif terhadap seluruh skenario *input* yang diujikan.

Secara keseluruhan, hasil interpretasi menunjukkan bahwa metode *Decision Table Testing* efektif dalam mendeteksi kesalahan logika pada validasi form. Penggunaan pendekatan eksplisit cocok digunakan pada form sederhana seperti Login dan Artikel dengan memberikan cakupan penuh terhadap kemungkinan *input*. Sementara itu, pendekatan implisit lebih sesuai digunakan pada form dengan kompleksitas tinggi karena dapat menjaga efisiensi pengujian tanpa mengurangi kualitas hasil. Dengan demikian, dapat disimpulkan bahwa sistem validasi pada Website Fishco telah berjalan sesuai dengan yang diharapkan, baik dari segi ketepatan logika maupun efisiensi pelaksanaan pengujian.

4.3. Analisis Efektifitas

Metode *Decision Table Testing* terbukti memberikan efisiensi dalam penyusunan *test case* karena penguji dapat langsung fokus pada kondisi yang relevan tanpa melakukan pengujian berulang. Jika dibandingkan dengan pengujian manual secara eksploratif, metode ini menghasilkan pendekatan yang lebih terstruktur, sistematis, dan efisien dalam mendeteksi kesalahan logika sistem. Penguji tidak hanya memperoleh pemahaman yang lebih jelas terhadap logika sistem, tetapi juga dapat menelusuri hubungan sebab-akibat antara *input* dan *output* dengan lebih mudah. Hal ini memungkinkan proses identifikasi kesalahan dilakukan dengan lebih cepat dan tepat. Selain itu, penerapan *Decision Table Testing* meningkatkan akurasi dalam mendeteksi kesalahan, khususnya pada tahap validasi *input*, di mana kesalahan logika atau kelalaian kondisi sering kali tidak teridentifikasi melalui metode pengujian manual. Secara keseluruhan, penerapan metode *Decision Table Testing* tidak hanya meningkatkan efektivitas pengujian perangkat lunak, tetapi juga berkontribusi terhadap kualitas dan keandalan sistem yang diuji. Pendekatan ini memberikan dasar analisis yang kuat dalam proses verifikasi dan validasi, sehingga sistem yang dihasilkan mampu berfungsi sesuai dengan kebutuhan dan harapan pengguna.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil pengujian dan analisis yang telah dilakukan, dapat disimpulkan bahwa penerapan metode *Decision Table Testing* pada pengujian sistem validasi form Website Fishco berhasil memastikan setiap fitur sistem berjalan sesuai dengan logika yang telah dirancang. Metode ini terbukti efektif dalam mendeteksi kesalahan logika serta meningkatkan efisiensi proses pengujian, khususnya pada validasi *input* form dengan tingkat kompleksitas yang berbeda. Pendekatan eksplisit menunjukkan hasil yang optimal pada form sederhana seperti Login dan Artikel, karena mampu memberikan cakupan pengujian penuh terhadap seluruh kombinasi *input*. Sementara itu, pendekatan implisit terbukti lebih efisien pada form yang memiliki kompleksitas tinggi seperti User, Penyakit Ikan, dan Produk, tanpa mengurangi ketelitian dalam proses pengujian. Hasil pengujian menunjukkan bahwa sistem mampu menolak *input* yang tidak sesuai dengan aturan validasi serta menerima data yang benar secara konsisten, tanpa ditemukan kesalahan pada proses validasi. Secara keseluruhan, metode *Decision Table Testing* mampu meningkatkan efektivitas dan efisiensi pengujian dengan menyediakan struktur pengujian yang sistematis, memperkuat akurasi deteksi kesalahan logika, serta memastikan bahwa Website Fishco berfungsi sesuai dengan kebutuhan dan harapan pengguna. Sebagai saran untuk pengembangan selanjutnya, pengujian dapat diperluas dengan menambahkan aspek non-fungsional, seperti keamanan lanjutan, performa sistem, dan

kompatibilitas lintas perangkat, agar kualitas sistem dapat dievaluasi secara lebih menyeluruh. Selain itu, pemanfaatan alat otomatisasi pengujian juga dapat dipertimbangkan untuk meningkatkan konsistensi serta efisiensi proses pengujian pada tahap pengembangan berikutnya.

DAFTAR PUSTAKA

- [1] K. Salsabila, F. T. Anggraeny, and A. M. Rizki, "Testing of the Decision Support System of Department Determination on High School Students Using Black Box Method Based on Equivalence Partitions," *Jurnal Informatika Polinema*, vol. 9, no. 1, pp. 39–44, Nov. 2022, doi: 10.33795/jip.v9i1.1062.
- [2] Riatun and E. D. Lestari, "Analysis of the Effect of Information Quality, System Quality, and Support Service Quality on User Satisfaction Levels and Its Implications for Blended E-Learning Continuance Intention to Use in the New Normal Era," *Formosa Journal of Sustainable Research*, vol. 1, no. 7, pp. 1067–1082, Dec. 2022, doi: 10.55927/fjsr.v1i7.2226.
- [3] S. W. Nurjihan *et al.*, "Analisis Decision Table Testing Untuk Pengujian Blackbox Website Pusat Studi Bencana IPB," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 11, no. 4, pp. 440–448, Dec. 2024, doi: <https://doi.org/10.35957/jatisi.v11i4.9618>.
- [4] F. Islamiah, R. Rusmiati, and R. Wijaya, "Penilaian Kepuasan Pengguna Website Sistem Informasi Akademik Menggunakan Metode Website Quality," *METIK JURNAL*, vol. 6, no. 2, pp. 133–139, Dec. 2022, doi: 10.47002/metik.v6i2.381.
- [5] G. I. Marthasari, A. T. Wahyuningsih, M. R. Aviansyah, M. A. Ramadhani, and Z. Rahmatullah, "Pengujian Website Infotech Menggunakan Teknik Black-Box Decision Table," *Jurnal Informatika Universitas Pamulang*, vol. 7, no. 1, pp. 115–119, Mar. 2022, doi: 10.32493/informatika.v7i1.17315.
- [6] S. J. Putri, D. G. P. Putri, and W. H. N. Putra, "Analisis Komparasi pada Teknik Black Box Testing (Studi Kasus: Website Lars)," *Journal of Internet and Software Engineering*, vol. 5, no. 1, pp. 23–28, May 2024, doi: 10.22146/jise.v5i1.9446.
- [7] I. S. Handayanto and I. Nuryasin, "Pengujian Blackbox Decision Table pada Sistem Aplikasi Mobile Sharing Story App," *Smart Comp: Jurnalnya Orang Pintar Komputer*, vol. 13, no. 2, Apr. 2024, doi: 10.30591/smartcomp.v13i2.6572.
- [8] Rahman Abdillah, Rudi Hermawan, Wawan Hermawansyah, Ibnu Adkha, and Heri Arifin, "Pengujian Perangkat Lunak Sistem Informasi Inventori pada Usaha Jasa Pengiriman Paket," *Polygon: Jurnal Ilmu Komputer dan Ilmu Pengetahuan Alam*, vol. 2, no. 4, pp. 166–175, Jul. 2024, doi: 10.62383/polygon.v2i4.199.
- [9] A. N. Hasibuan and T. Dirgahayu, "Pengujian dengan Unit Testing dan Test case pada Proyek Pengembangan Modul Manajemen Pengguna," *Automata (Ajang Unjuk Tugas Akhir oleh Mahasiswa Informatika)*, Jan. 2021.
- [10] B. Hardika *et al.*, "Pengujian Blackbox Testing Website Garuda Farm Menggunakan Teknik Equivalence Partitioning," *JURNAL KRIDATAMA SAINS DAN TEKNOLOGI*, vol. 6, no. 02, pp. 740–753, Dec. 2024, doi: 10.53863/kst.v6i02.1420.
- [11] N. M. D. Febriyanti, A. A. K. O. Sudana, and I. N. Piarsa, "Implementasi Black Box Testing pada Sistem Informasi Manajemen Dosen," *Jurnal Ilmiah Teknologi dan Komputer*, vol. 2, no. 3, Dec. 2021.
- [12] Muhammad Jibril, Zulrahmadi, and 3Muhammad Amin, "Pengujian Sistem Informasi E-Modul pada SMPN 1 Tempuling Menggunakan Black Box Testing," *JURNAL PERANGKAT LUNAK*, vol. 6, no. 2, pp. 327–332, Jun. 2024, doi: 10.32520/jupel.v6i2.3326.
- [13] Uminingsih, M. Nur Ichsanudin, M. Yusuf, and S. Suraya, "Pengujian Fungsional Perangkat Lunak Sistem Informasi Perpustakaan dengan Metode Black Box Testing bagi Pemula," *STORAGE: Jurnal Ilmiah Teknik dan Ilmu Komputer*, vol. 1, no. 2, pp. 1–8, May 2022, doi: 10.55123/storage.v1i2.270.
- [14] S. S. Fatihatur, S. E. Safitri, N. Rahmadani, I. K. Anshor, and D. Arsy S, "Pengujian Website Sistem Pakar Penentuan Kepribadian Menggunakan Metode Forward Chaining dan Decision Table," *remik*, vol. 9, no. 1, pp. 48–59, Jan. 2025, doi: 10.33395/remik.v9i1.14286.
- [15] J. Shadiq, A. Safei, and R. W. R. Loly, "Pengujian Aplikasi Peminjaman Kendaraan Operasional Kantor Menggunakan BlackBox Testing," *INFORMATION MANAGEMENT FOR EDUCATORS AND PROFESSIONALS: Journal of Information Management*, vol. 5, no. 2, p. 97, Jul. 2021, doi: 10.51211/imbi.v5i2.1561.
- [16] F. D. Hartono and Y. Sugiarti, "Perbandingan Metode Equivalence Partitions dan Boundary Value Analysis pada Pengujian Black Box," *Majalah Ilmiah METHODODA*, vol. 12, no. 2, pp. 153–159, Aug. 2022, doi: 10.46880/methoda.Vol12No2.pp153-159.
- [17] B. Pietrzak, *Software Testing and Change Management: Test Models — Combinational Models*. Poznan University of Technology