

PENGUJIAN PADA WEBSITE SMARTPETSCARE UNTUK LAYANAN GROOMING HEWAN MENGGUNAKAN METODE BLACK BOX TESTING

Nur Rahma Ditta Zahra, Anatasya Wenita Putri, Kanaya Sabila Azzahra, David Reza Widhiwipati, Muhammad Ilham Nurfajri, Gema Parasti Mindara, Aditya Wicaksono

Teknologi Rekayasa Perangkat Lunak, Teknologi Rekayasa Komputer, IPB University

Jl. Kumbang No.14, RT.02/RW.06, Bogor - Jawa Barat 16128

gemaparasti@apps.ipb.ac.id

ABSTRAK

Penelitian ini bertujuan untuk menguji fungsionalitas sistem berbasis web SmartPetsCare, yang dirancang untuk mendukung layanan grooming hewan peliharaan secara online. Website ini menyediakan berbagai fitur, seperti pendaftaran hewan peliharaan, pengaturan jadwal layanan, pelacakan riwayat grooming, serta akses informasi melalui artikel, tips perawatan, dan showcase produk. Pendekatan Black Box Testing digunakan untuk menguji fungsi-fungsi utama sistem berdasarkan spesifikasi kebutuhan pengguna, tanpa mengevaluasi struktur internal atau kode programnya. Pengujian dilakukan pada fitur login, registrasi, serta operasi CRUD (*Create, Read, Update, Delete*) untuk memastikan setiap fungsi berjalan sesuai harapan. Hasil pengujian menunjukkan bahwa fitur-fitur utama website beroperasi sesuai spesifikasi yang ditentukan. Semua skenario pengujian berhasil memberikan hasil yang diharapkan, meskipun terdapat beberapa peluang perbaikan, seperti optimalisasi antarmuka pengguna dan pengayaan fitur tambahan untuk meningkatkan kualitas pengalaman pengguna. Penelitian ini diharapkan dapat menjadi referensi dalam pengujian dan pengembangan perangkat lunak berbasis web, khususnya dengan penerapan metode Black Box Testing untuk memastikan keandalan sistem.

Kata kunci : *Black Box Testing, Sistem Informasi Grooming, SmartPetsCare, Pengujian Perangkat Lunak, CRUD*

1. PENDAHULUAN

Uji coba perangkat lunak adalah proses menjalankan aplikasi untuk mendeteksi adanya cacat atau kesalahan. Keberhasilan dalam pengujian tercapai apabila tes tersebut berhasil mengungkapkan masalah yang tidak ditemukan dalam pemeriksaan sebelumnya. Uji perangkat lunak memainkan peran yang sangat penting dalam sebuah sistem informasi, karena melalui proses ini, kesalahan atau masalah pada perangkat lunak dapat terdeteksi. Kesalahan atau bug dalam perangkat lunak dapat mengganggu kelancaran proses bisnis yang seharusnya didukung oleh aplikasi tersebut, bahkan mengharuskan perbaikan atau pengembangan ulang jika jumlahnya terlalu banyak. Oleh karena itu, dengan dilakukan pengujian, diharapkan kesalahan atau kekurangan dalam aplikasi dapat dikurangi, sekaligus menjadi indikator untuk menilai kualitas dari perangkat lunak tersebut [1].

Website SmartPetsCare dirancang sebagai sistem informasi berbasis web untuk layanan grooming hewan peliharaan. Fitur utama platform ini meliputi pendaftaran hewan, penjadwalan, pemantauan riwayat grooming, serta penyediaan artikel, tips grooming, dan showcase produk berkualitas. Sistem ini tidak memiliki fitur jual-beli, tetapi fokus pada memberikan layanan yang optimal untuk pemilik hewan. Dengan antarmuka yang ramah pengguna, SmartPetsCare bertujuan untuk memberikan pengalaman yang mudah dan menyenangkan bagi para pemilik hewan peliharaan dalam merawat dan memastikan kesehatan hewan mereka. Selain itu, platform ini juga menyediakan informasi yang berguna untuk membantu pemilik membuat keputusan yang tepat dalam merawat hewan peliharaan mereka..

Black box testing adalah metode pengujian perangkat lunak yang menilai kinerja sistem berdasarkan fungsionalitasnya, tanpa mempertimbangkan struktur internalnya. Pengujian dilakukan dengan mengamati input yang diberikan dan output yang dihasilkan, tanpa perlu memahami kode atau algoritma yang mendasarinya [2].

Metode ini mengambil sudut pandang pengguna akhir, sehingga potensi masalah atau kekurangan dalam sistem dapat teridentifikasi dan diperbaiki untuk meningkatkan kualitasnya [3].

Pendekatan ini relevan karena menguji fungsi-fungsi sistem dari perspektif pengguna tanpa memerlukan akses ke kode sumber [4].

Penelitian ini bertujuan untuk mengevaluasi keandalan fitur pada situs web SmartPetsCare dan memberikan rekomendasi perbaikan. Selain itu, hasil penelitian ini diharapkan dapat menjadi referensi untuk pengembangan lebih lanjut dalam pengujian perangkat lunak berbasis web.

2. TINJAUAN PUSTAKA

2.1. Pengujian

Pengujian sistem perangkat lunak merupakan tahapan untuk menjalankan dan menguji aplikasi yang telah dikembangkan oleh tim pengembang, dengan tujuan untuk mendeteksi masalah atau kesalahan (seperti bug) yang mungkin ada. Tujuan utama dari pengujian ini adalah untuk memastikan bahwa aplikasi bekerja sesuai dengan yang diharapkan dan memiliki kualitas yang baik. Pengujian sangat penting karena dalam proses pengembangan perangkat lunak, sering kali ada aspek yang terlewat atau tidak sempurna, sehingga pengujian menjadi langkah krusial untuk

menemukan dan memperbaiki masalah tersebut [5]. Penelitian sebelumnya oleh Yuliawati et al. (2022) menunjukkan bahwa metode Black Box Testing efektif untuk mengevaluasi fungsionalitas sistem, khususnya pada aplikasi berbasis web yang melibatkan interaksi langsung dengan pengguna [5].

2.2. Black Box Testing

Black-Box Testing adalah teknik evaluasi perangkat lunak yang lebih mengutamakan pengujian terhadap fungsi-fungsi yang ada, tanpa melibatkan analisis mendalam terhadap struktur internal atau kode sumber perangkat lunak tersebut. Fokus utama dari pengujian ini adalah untuk memastikan bahwa setiap fungsi yang ada bekerja sesuai dengan spesifikasi yang telah ditentukan, serta memastikan bahwa input yang diberikan menghasilkan output yang tepat seperti yang diharapkan [5].

Pengujian ini dilakukan dengan pendekatan yang berfokus pada pengalaman pengguna akhir (*end-user*), dimana pengujian dilakukan berdasarkan interaksi pengguna dengan antarmuka dan menu yang tersedia dalam aplikasi [6].

Penelitian sebelumnya juga menunjukkan keberhasilan penerapan berbagai teknik dalam Black Box Testing. Rahadi dan Vikasari (2020) menggunakan metode Partisi Ekuivalen untuk menguji aplikasi manajemen barang dan berhasil mendeteksi bug secara efisien dengan mengurangi jumlah pengujian yang diperlukan [9].

Sementara itu, Andika dan Syam (2020) menerapkan pengujian transisi status untuk memastikan stabilitas sistem informasi kesehatan berbasis web, yang memungkinkan deteksi kesalahan pada perubahan status sistem [8].

2.3. Jenis-Jenis Pengujian dalam Black Box Testing

Terdapat berbagai pendekatan yang digunakan dalam *Black Box Testing*, masing-masing dengan tujuan dan metode yang berbeda untuk menguji kinerja sistem. Berikut adalah beberapa jenis pengujian yang umum diterapkan dalam metode ini:

2.3.1. Partisi Ekuivalen (Equivalence Partitioning)

Teknik ini membagi data input menjadi beberapa kelompok atau kategori yang dianggap memiliki karakteristik serupa. Dengan demikian, pengujian dapat difokuskan pada satu kasus per kelompok, yang dapat mewakili seluruh kelompok tersebut, sehingga mengurangi jumlah pengujian yang diperlukan.

2.3.2. Analisis Nilai Batas (Boundary Value Analysis)

Pendekatan ini mengidentifikasi kesalahan yang mungkin muncul pada batas nilai yang diterima oleh sistem, baik diluar maupun di dalam rentang yang diizinkan. Pengujian ini bertujuan untuk memastikan sistem dapat menangani nilai batas dengan benar, baik itu nilai minimum maupun maksimum yang valid.

a. Fuzzing

Fuzzing adalah teknik Untuk mengidentifikasi kesalahan atau masalah dalam sistem perangkat lunak dengan cara menyuntikkan data yang tidak valid atau cacat. Tujuannya adalah untuk menguji seberapa tahan sistem terhadap input yang terbilang ekstrem atau tidak terduga, yang dapat menyebabkan kegagalan atau kerusakan pada aplikasi.

b. Grafik Penyebab-Efek (Cause-Effect Graphing)

Dalam pendekatan ini, digunakan grafik untuk menggambarkan hubungan antara penyebab dan efek dalam sistem. Teknik ini memungkinkan pengujian dilakukan berdasarkan interaksi antara input dan output, serta memvisualisasikan bagaimana perubahan dalam satu elemen dapat mempengaruhi keseluruhan sistem.

c. Orthogonal Array Testing

Pengujian ini diterapkan ketika ruang lingkup input tergolong kecil, namun tetap kompleks untuk diuji secara menyeluruh. Dengan memilih kombinasi input yang relevan dan teroptimalkan, teknik ini membantu mengurangi jumlah pengujian yang diperlukan, meskipun skala pengujian cukup besar.

d. All Pair Testing

Teknik ini merancang test case yang menguji semua pasangan kombinasi input yang mungkin. Pengujian ini memastikan bahwa setiap pasangan input diuji secara menyeluruh, untuk mengidentifikasi potensi masalah yang mungkin timbul dari kombinasi input yang berbeda.

e. State Transition Testing

Pendekatan ini digunakan untuk menguji perubahan status atau kondisi sistem, terutama pada aplikasi yang memiliki status yang saling berhubungan. Dengan menggambarkan perubahan kondisi dalam bentuk grafik, teknik ini membantu memverifikasi bagaimana sistem bereaksi terhadap peralihan antar status [7].

2.4. Sistem Informasi Berbasis Web

Sistem informasi berbasis web adalah sebuah struktur yang terdiri dari berbagai elemen yang saling terhubung dan berfungsi untuk mengumpulkan, mengelola, menyimpan, serta mentransfer data dalam berbagai bentuk, seperti teks, gambar, suara, dan informasi lainnya, yang disajikan dalam format hypertext. Tujuan utama dari sistem ini adalah untuk menyediakan akses informasi secara online, mendukung berbagai proses dalam organisasi agar dapat mencapai tujuannya. Untuk menampilkan halaman-halaman dalam bentuk hypertext kepada pengguna, web browser menggunakan aplikasi khusus yang berfungsi sebagai penghubung antara pengguna dan server, yang berkomunikasi menggunakan protokol yang dikenal sebagai HTTP (*Hypertext Transfer Protocol*) [8].

Sistem informasi berbasis web memiliki kemampuan untuk menyediakan akses data real-time dan mendukung pengelolaan informasi yang efisien. Ramadhan dan Purwandari (2018) menyatakan bahwa

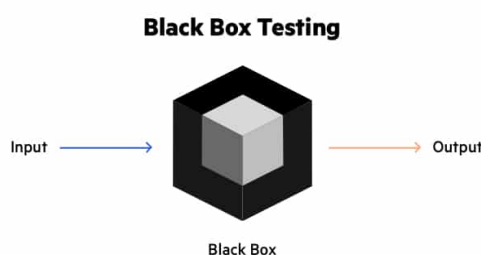
fleksibilitas sistem ini membuatnya ideal untuk berbagai kebutuhan organisasi [6].

Penelitian Ichsanudin et al. (2022) menekankan bahwa penerapan Black Box Testing pada sistem informasi berbasis web dapat mengidentifikasi kesalahan dengan efektif, meningkatkan akurasi data yang ditampilkan kepada pengguna [7].

3. METODE PENELITIAN

Penelitian ini mengadopsi metode Black Box Testing, yang berfokus pada pengujian terhadap fungsionalitas sistem. Dalam pengujian ini, yang sering disebut sebagai pengujian berbasis perilaku, penguji tidak perlu memahami struktur internal atau logika yang ada di dalam perangkat lunak yang diuji. Pengujian didasarkan pada spesifikasi kebutuhan yang telah ditetapkan, tanpa melibatkan analisis kode. Dengan demikian, pengujian dilakukan dari sudut pandang pengguna akhir untuk memastikan apakah perangkat lunak berfungsi sesuai dengan yang diharapkan [10].

Berbagai teknik dapat digunakan dalam Black Box Testing, antara lain pembagian partisi, analisis batas nilai, diagram sebab-akibat, orthogonal array testing, pengujian transisi status, dan fuzzing. Selain itu, Black Box Testing memiliki berbagai keuntungan dan kelemahan. Salah satu keuntungan utamanya adalah kemampuannya untuk menemukan aspek-aspek yang mungkin tidak sesuai atau tidak tercakup dalam spesifikasi kebutuhan perangkat lunak. Namun, kekurangan utama dari pengujian ini adalah keterbatasan pengetahuan penguji mengenai perangkat lunak yang diuji, yang dapat menyebabkan pengujian tidak dapat mencakup seluruh bagian dari sistem yang diuji [9].



Gambar 1. Black box testing

Proses pengujian dimulai dengan identifikasi fitur utama sistem yang akan diuji, yaitu kemampuan CRUD (Create, Read, Update, Delete) pada menu, produk, dan layanan. Langkah pertama adalah menyusun skenario pengujian berdasarkan spesifikasi yang telah ditetapkan. Skenario ini mencakup berbagai kondisi input, termasuk input valid, tidak valid, dan nilai-nilai ekstrim. Setiap skenario dirancang untuk

mengevaluasi bagaimana sistem merespons berbagai jenis masukan dan memastikan bahwa hasil keluaran sesuai dengan kebutuhan yang diharapkan.

Pengujian dilakukan secara manual dengan memberikan input yang dirancang sebelumnya ke dalam sistem dan mencatat respons sistem. Hasil yang diperoleh selanjutnya dibandingkan dengan hasil yang seharusnya didapatkan untuk menentukan keberhasilan pengujian. Semua hasil pengujian didokumentasikan, mencakup status setiap skenario pengujian (berhasil atau gagal) dan catatan mengenai anomali atau bug yang ditemukan.

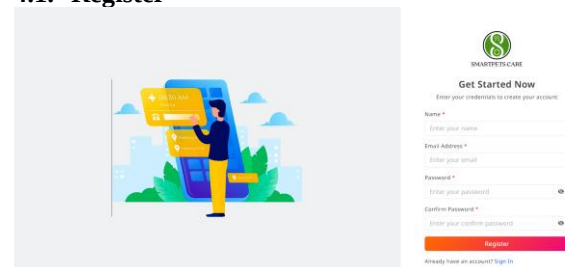
Setelah pengujian selesai, hasilnya dianalisis untuk mengevaluasi tingkat keandalan sistem dan menemukan area yang memerlukan perbaikan. Analisis ini dilakukan dengan memeriksa apakah fitur sistem telah memenuhi spesifikasi yang ditentukan. Dokumentasi hasil pengujian disusun secara sistematis untuk memberikan gambaran yang jelas tentang performa sistem dan memberikan rekomendasi perbaikan bagi pengembangan lebih lanjut.

Metode ini dirancang untuk memberikan hasil yang komprehensif, memastikan bahwa semua fitur sistem SmartPetsCare berjalan sesuai spesifikasi, dan memberikan masukan yang bermanfaat untuk peningkatan kualitas layanan.

4. HASIL DAN PEMBAHASAN

Hasil pengujian sistem SmartPetsCare menggunakan metode *Black Box Testing* menunjukkan bahwa fitur login, register, serta fungsi CRUD (*Create, Read, Update, Delete*) pada produk dan artikel telah diuji dan menghasilkan output yang diharapkan. Berikut adalah ringkasan hasil pengujian.

4.1. Register



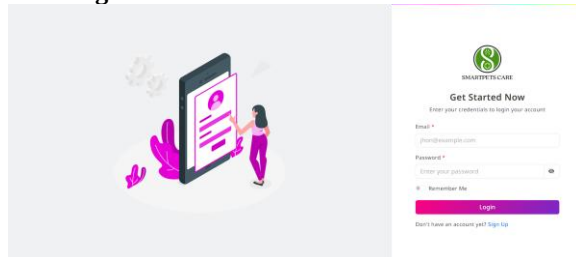
Gambar 2. Tampilan register

Fitur register pada website SmartPetsCare memungkinkan pengguna baru membuat akun dengan mengisi nama, email, password, dan confirm password. Sistem akan memvalidasi data input, memastikan semua field terisi, email dalam format yang benar, dan password cocok dengan confirm password. Jika valid, data pengguna disimpan dan pengguna diarahkan ke halaman login.

Tabel 1. Pengujian register

No	Skenario	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menginputkan nama, email, password, dan confirm password dengan benar , lalu klik “Register”	Nama: xxx Email: xxx Password: xxx Password: xxx	Akun berhasil dibuat, diarahkan ke halaman login.	Sesuai harapan	Valid
2	Mengosongkan semua isi field register, lalu klik “Register”	Nama: - Email: - Password: - Password: -	Sistem menampilkan pesan error "Field wajib harus diisi".	Sesuai harapan	Valid
3	Menginputkan semua field dengan password dan confirm password yang berbeda, lalu klik “Register”	Nama: xxx Email: xxx Password: xxx Password: xxx	Sistem menampilkan pesan error "Password tidak cocok".	Sesuai harapan	Valid

4.2. Login



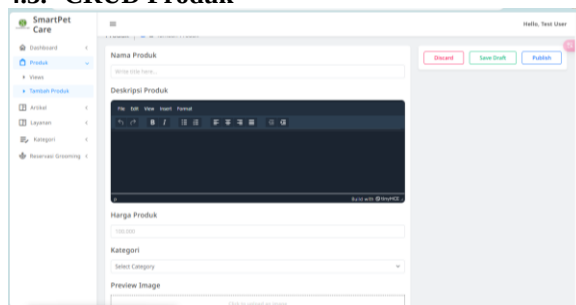
Gambar 3. Tampilan login

Fitur login digunakan untuk mengautentikasi pengguna dengan email dan password. Sistem akan memverifikasi kecocokan data input dengan informasi yang tersimpan dalam basis data. Jika valid, pengguna akan diarahkan ke dashboard untuk mengakses fitur yang tersedia.

Tabel 2. Pengujian Login

No	Skenario	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menginputkan username dan password dengan benar , lalu klik “Login”	Nama: xxx Password: xxx	Berhasil login, diarahkan ke dashboard Admin	Sesuai harapan	Valid
2	Menginputkan username yang salah dan password yang benar, lalu klik “Login”	Nama: xxx Password: xxx	Sistem menampilkan pesan error "Username atau password salah".	Sesuai harapan	Valid
3	Mengosongkan semua isi field login, lalu klik “Login”	Nama: - Password: -	Sistem menampilkan pesan error "Field wajib harus diisi".	Sesuai harapan	Valid

4.3. CRUD Produk



Gambar 4. Tampilan crud produk

data produk. Dengan fitur *Create*, admin dapat menambahkan produk baru beserta detail seperti deskripsi, harga, dan status ketersediaan. *Read* memungkinkan admin untuk melihat dan memverifikasi informasi produk yang telah ada, memastikan bahwa data yang ditampilkan akurat. Melalui fitur *Update*, admin dapat mengubah informasi produk, seperti harga atau deskripsi, sesuai kebutuhan. Terakhir, fitur *Delete* memberi kemampuan kepada admin untuk menghapus produk yang tidak lagi relevan, memastikan bahwa daftar produk tetap terjaga dengan baik.

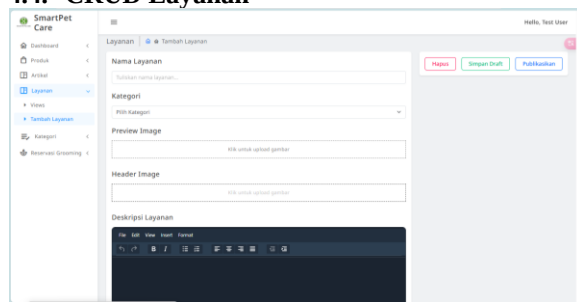
Fitur CRUD pada produk di website SmartPetsCare memudahkan admin dalam mengelola

Tabel 3. Pengujian CRUD produk

No	Skenario	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Admin di halaman produk dan klik tambah produk serta menginputkan semua field produk dengan benar. Kemudian, klik “Simpan”	Input data produk lengkap: Nama Produk, Deskripsi, Harga, Stok. Klik "Simpan"	Data produk tersimpan dan ditampilkan di halaman produk	Sesuai harapan	Valid

2	Admin di halaman produk dan klik tambah produk serta menginputkan field produk dengan data yang tidak lengkap. Kemudian, klik "Simpan"	Input data produk tidak lengkap (misal, tanpa harga atau deskripsi). Klik "Simpan"	Sistem menampilkan pesan error "Field wajib harus diisi"	Sesuai harapan	Valid
3	User berada di halaman produk dan admin sudah menambahkan produk. Lalu, user melihat produk yang ditambahkan admin	User membuka halaman produk dan memeriksa daftar produk	Semua data layanan ditampilkan dengan benar	Sesuai harapan	Valid
4	Admin di halaman produk dan klik ubah produk serta menginputkan field produk dengan lengkap. Kemudian, klik "Simpan"	Admin memilih produk yang ada, mengubah data produk (misalnya, nama atau harga), lalu klik "Simpan"	Data produk diperbarui sesuai input baru	Sesuai harapan	Valid
5	Admin di halaman produk dan klik ubah produk serta menginputkan field produk dengan data tidak lengkap. Kemudian, klik "Simpan"	Admin memilih produk yang ada, mengubah data produk dengan input yang tidak lengkap, lalu klik "Simpan"	Sistem menampilkan pesan error "Field wajib harus diisi"	Sesuai harapan	Valid
6	Admin di halaman produk dan memilih produk yang ingin dihapus, kemudian klik "Hapus"	Admin memilih produk yang ingin dihapus dan mengklik "Hapus"	Data produk dihapus dan hilang dari halaman produk	Sesuai harapan	Valid

4.4. CRUD Layanan



Gambar 5. Tampilan crud layanan

Fitur CRUD pada layanan di website SmartPetsCare memungkinkan admin untuk mengelola informasi layanan grooming dengan mudah. Fitur *Create* memungkinkan admin menambahkan layanan baru, seperti jenis grooming, harga, dan deskripsi layanan. Dengan fitur *Read*, admin dapat melihat dan memastikan bahwa informasi layanan yang ada sudah akurat. Fitur *Update* memungkinkan admin untuk memperbarui detail layanan, seperti harga atau deskripsi, jika ada perubahan. Terakhir, fitur *Delete* memungkinkan admin untuk menghapus layanan yang tidak lagi ditawarkan.

Tabel 4. Pengujian CRUD layanan

No	Skenario	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Admin di halaman layanan dan klik tambah layanan serta menginputkan semua field layanan dengan benar. Kemudian, klik "Simpan"	Input data layanan lengkap: Nama Layanan, Deskripsi, Harga. Klik "Simpan"	Data layanan tersimpan dan ditampilkan di halaman layanan	Sesuai harapan	Valid
2	Admin di halaman layanan dan klik tambah layanan serta menginputkan field layanan dengan data yang tidak lengkap. Kemudian, klik "Simpan"	Input data layanan tidak lengkap (misal, tanpa deskripsi atau harga). Klik "Simpan"	Sistem menampilkan pesan error "Field wajib harus diisi"	Sesuai harapan	Valid
3	User berada di halaman layanan dan admin sudah menambahkan layanan. Lalu, user melihat produk yang ditambahkan admin	User membuka halaman layanan dan memeriksa daftar layanan	Semua data layanan ditampilkan dengan benar	Sesuai harapan	Valid
4	Admin di halaman layanan dan klik ubah layanan serta menginputkan field layanan dengan lengkap. Kemudian, klik "Simpan"	Admin memilih layanan yang ada, mengubah data layanan (misalnya, nama atau harga), lalu klik "Simpan"	Data produk diperbarui sesuai input baru	Sesuai harapan	Valid
5	Admin di halaman layanan dan klik ubah layanan serta menginputkan field layanan dengan data tidak lengkap. Kemudian, klik "Simpan"	Admin memilih layanan yang ada, mengubah data layanan dengan input yang tidak lengkap, lalu klik "Simpan"	Sistem menampilkan pesan error "Field wajib harus diisi"	Sesuai harapan	Valid

No	Skenario	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
6	Admin di halaman layanan dan memilih layanan yang ingin dihapus, kemudian klik "Hapus"	Admin memilih layanan yang ingin dihapus dan mengklik "Hapus"	Data layanan dihapus dan hilang dari halaman layanan	Sesuai harapan	Valid

5. KESIMPULAN DAN SARAN

Berdasarkan hasil pengujian sistem SmartPetsCare menggunakan metode Black Box Testing, dapat disimpulkan bahwa fitur login, register, serta fungsi CRUD (*Create, Read, Update, Delete*) pada produk dan layanan berjalan sesuai dengan spesifikasi yang ditetapkan, dengan sistem dapat menangani input valid maupun tidak valid dengan baik. Meskipun hasil pengujian menunjukkan kinerja yang memadai, masih ada kesempatan untuk perbaikan, khususnya dalam meningkatkan antarmuka pengguna untuk pengalaman yang lebih optimal. Oleh karena itu, untuk penelitian selanjutnya disarankan untuk melakukan pengujian lebih lanjut pada aspek keamanan, pengujian beban, dan integrasi sistem, serta memperdalam pengujian pada fitur tambahan yang belum diuji, seperti manajemen akun pengguna dan notifikasi, untuk memastikan keandalan dan stabilitas sistem secara menyeluruh.

DAFTAR PUSTAKA

- [1] D. Ahrizal, M. K. Miftah, R. Kurniawan, T. Zaelani, and Y. Yulianti, "Pengujian perangkat lunak sistem informasi peminjaman PlayStation dengan teknik Boundary Value Analysis menggunakan metode Black Box Testing," *Jurnal Informatika Universitas Pamulang*, vol. 5, no. 1, pp. 73-77, 2020.
- [2] D. Yulawati, A. Andriyadi, and N. Nursiyanto, "Pengujian sistem informasi E-Monitoring pengelolaan pembangunan desa dengan menggunakan metode Blackbox Testing," *TEKNIKA: Jurnal Ilmiah Bidang Ilmu Rekayasa*, vol. 16, no. 2, pp. 303-310, Nov. 2022, doi: 10.5281/ZENODO.7535944.
- [3] D. Wintana, D. Pribadi, and M. Y. Nurhadi, "Analisis perbandingan efektivitas White-Box Testing dan Black-Box Testing," *Jurnal Larik: Ladang Artikel Ilmu Komputer*, vol. 2, no. 1, pp. 8-16, Jul. 2022. [Online]. Available: <http://103.75.24.116/index.php/larik/article/view/1382>.
- [4] M. E. Khan, "Different approaches to Black Box Testing technique for finding errors," Jul. 21, 2021. [Online]. Available: <https://papers.ssrn.com/abstract=3890672>. Accessed: Nov 26, 2024.
- [5] W. N. Cholifah, Y. Yulianingsih, and S. M. Sagita, "Pengujian Black Box Testing pada aplikasi action & strategy berbasis Android dengan teknologi PhoneGap," *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, vol. 3, no. 2, pp. 206-210, 2018.
- [6] F. Ramadhan and N. Purwandari, "Sistem informasi penjualan berbasis web pada PT. Mustika Jati," *Jurnal Sains dan Teknologi*, vol. 5, no. 1, pp. 43-57, 2018.
- [7] M. N. 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, 2022.
- [8] O. Andika and S. Syam, "Sistem informasi pelayanan kesehatan berbasis web di Klinik Mutiara Bunda Pasar Kemis Kabupaten Tangerang," *Jurnal Ilmiah Fakultas Teknik*, vol. 1, no. 2, pp. 74-79, 2020.
- [9] A. C. Praniffa, A. . Syahri, F. Sandes, U. Fariha, Q. A. Giansyah, and M. Hamzah, "Pengujian Sistem Informasi Parkir Berbasis Web Pada UIN SUSKA RIAU Menggunakan White Box dan Black Box Testing," *JTISI*, vol. 1, no. 1, pp. 1-16, Mar. 2023.
- [10] Rahadi, N. W., and C. Vikasari, "Pengujian Software Aplikasi Perawatan Barang Milik Negara Menggunakan Metode Black Box Testing Equivalence Partitions," *Jurnal Infotekmesin*, vol. 11, no. 01, pp. 57-61, 2020