



Pengujian *Black Box* Pada Aplikasi *Access By KAI* Menggunakan Teknik *Equivalence Partitioning*

Puja Vita Maharani¹, Vina Zahrotun Nazah^{2*}, Maratus Sholiha³, Alif Haikal⁴

^{1,2,3,4}Informatika, Teknik Informatika, Universitas Bina Insani, Kota Bekasi, Indonesia

Email: ¹pujavitamaharani@gmail.com, ^{2*}vinazahrotunnazah@gmail.com, ³maratus204@gmail.com,

⁴haikalsialip999@gmail.com

(* : coressponding author)

Abstrak – Aplikasi *Access by KAI*, merupakan aplikasi yang dikembangkan oleh PT Kereta Api Indonesia untuk mempermudah dalam melakukan pemesanan tiket kereta secara online. Untuk dapat memastikan aplikasi memenuhi kebutuhan pengguna dan berfungsi sesuai harapan dilakukan pengujian menggunakan metode *Black Box Testing* dengan Teknik *Equivalence Partitioning*. Pengujian ini melibatkan pengujian fungsi utama pada halaman login, halaman data diri, dan halaman kereta. Pengujian alfa dilakukan secara internal untuk mendeteksi dan memperbaiki bug. Hasil pengujian alpha mendeteksi adanya error atau bug yang terjadi pada menu data diri yang memerlukan perbaikan lebih lanjut. Penelitian ini menunjukkan bahwa metode *Equivalence Partitioning* efektif dalam mengidentifikasi kesalahan dan memastikan bahwa aplikasi berfungsi dengan baik dalam berbagai situasi

Kata Kunci Aplikasi *Access by KAI*, *Black Box Testing*, Teknik *Equivalence Partitioning*, Pengujian Alfa

Abstract – *The Access by KAI application, is an application developed by PT Kereta Api Indonesia to make it easier to order train tickets online. To be able to ensure the application meets user needs and functions as expected, testing is carried out using the Black Box Testing method with the Equivalence Partitioning Technique. This test involves testing the main functions on the login page, personal data page, and train page. Alpha testing is done internally to detect and fix bugs. The alpha test results detect errors or bugs that occur in the personal data menu that require further improvement. This research shows that the Equivalence Partitioning method is effective in identifying errors and ensuring that the application functions properly in various situations.*

Keywords: *Access by KAI Application, Black Box Testing, Equivalence Partitioning Technique, Alfa Testing*

1. PENDAHULUAN

Pengujian pada sebuah perangkat lunak adalah proses menguji dan memeriksa bahwa perangkat lunak atau aplikasi melakukan apa yang diminta sesuai dengan kebutuhan (Ismail et al., 2023). Dalam proses pengembangan perangkat lunak masalah yang sering ditemukan adalah bug dan error baik pada sintak ataupun

Metode *Black Box Testing* merupakan pengujian yang berfokus pada fungsionalitas suatu perangkat lunak, dimana penguji dapat mendefinisikan kumpulan kondisi masukan dan melakukan pengujian pada fungsionalitas program. *Black Box Testing* merupakan pengujian yang dilakukan dengan tidak mengacu pada struktur internal dari komponen ataupun sebuah sistem. Sedangkan menurut (Putri, 2022) metode pengujian *Black Box* merupakan pengujian dengan fokus pada fungsionalitas suatu perangkat lunak.

Dalam strategi pengujian *Black Box* memiliki beberapa teknik, teknik yang kami gunakan dalam pengujian Aplikasi *Access by KAI* adalah Teknik *equivalence partitions*. Teknik *Equivalence Partitioning* merupakan sebuah teknik pengujian yang dilakukan berdasarkan masukan data pada tiap form yang ada pada sistem aplikasi. Teknik *equivalence partitioning* memecah atau membagi domain *input* dari program ke dalam kelas-kelas data sehingga *test case* dapat diperoleh (Haryanto et al., 2023). Sebagai contoh dalam pengembangan aplikasi seperti *Access by KAI* untuk kebutuhan transportasi publik, metode ini dapat digunakan untuk memastikan kinerja dan keandalan aplikasi dalam berbagai situasi penggunaan.

Penelitian ini bertujuan untuk memastikan semua fungsi pada halaman *login*, halaman data diri dan halaman kereta di Aplikasi *Access by KAI* berjalan tanpa adanya bug atau *error*. Hal ini penting untuk memastikan bahwa aplikasi memenuhi standar kelayakan perangkat lunak dan memberikan layanan yang optimal serta aman bagi para penggunanya. Pengujian ini difokuskan



pada halaman login, halaman data diri, dan halaman kereta pada aplikasi *Access by KAI*. Pada halaman kereta, pengujian mencakup halaman antar kota, kereta lokal, kereta *commuter line*, dan kereta *whoosh*. Rincian pengujian meliputi validasi data *login*, input dan pembaruan data pribadi, pencarian jadwal kereta, pemesanan dan pembayaran kereta serta konfirmasi tiket kereta. Pengujian ini nantinya diharapkan dapat memastikan bahwa semua fungsi utama pada halaman halaman tersebut berjalan dengan baik tanpa adanya bug atau *error*.

2. METODOLOGI PENELITIAN

2.1 *Black Box Testing*

Black box testing merupakan pengujian yang menguji hanya dibagian *design*, fungsi aplikasi dan kegunaan aplikasinya. Pada *black box testing* pengujian tidak perlu melihat sampai kedalam program atau sintaknya seperti pada pengujian *white box testing*. Pada *black box testing* terdapat beberapa teknik pengujian diantaranya adalah *Equivalence Partitioning*, *Fuzzing*, *Boundary Value Analysis*, *Cause-Effect Graph*, *Orthogonal Array Testing*, *State Transition*, dan *All Pair Transition*(Amalia et al., 2021)

2.2 Teknik *Equivalence Partitions*

Teknik Equivalence Partitioning merupakan salah satu teknik *black box testing*. Teknik ini berfokus pada nilai *input* dan *output* yang dihasilkan pada suatu fungsi perangkat lunak. Teknik ini mencoba untuk mendefinisikan kasus uji yang mengungkap kelas kesalahan, sehingga mengurangi jumlah kasus uji yang harus dikembangkan.(Muhammad Arofiq et al., 2023)

2.3 Pengujian Alfa

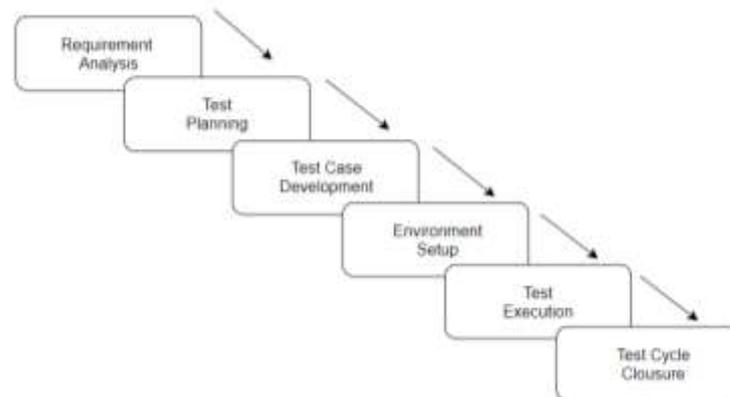
Alfa testing merupakan tahap pengujian yang pertama kali dilakukan ketika sebuah produk dikembangkan. *Alfa testing* adalah pengujian yang bertujuan untuk memastikan bahwa aplikasi yang diuji dapat berjalan dengan lancar tanpa gangguan *error* atau bug (Achmad & Yulfitri, 2020). Pengujian alfa digunakan untuk memverifikasi status kualitas dari sebuah perangkat lunak dalam waktu yang ditentukan.

2.4 Tujuan Pengujian Alfa

Pengujian alfa adalah pengujian dengan cara melakukan sebuah test pada aplikasi dengan menggunakan test case. Pengujian *alfa* sering digunakan untuk perangkat lunak sebagai bentuk pengujian penerimaan internal sebelum perangkat lunak masuk ke tahap pengujian beta. Pada pengujian aplikasi *Access by KAI* tim pengujian melakukan tes satu persatu dengan menginput data dengan format yang benar dan salah. Pengujian ini dibatasi hanya pada halaman registrasi, halaman login, halaman data diri dan halaman kereta yang memuat halaman kereta antar kota, kereta lokal, kereta *commuter line*, dan kereta *whoosh*. Tujuan dari pengujian alfa sendiri yaitu untuk memastikan bahwa Aplikasi *Access by KAI* dapat berjalan dengan baik dan akan membantu proses pengembangan kedepannya.

2.5 Proses dan Metodologi

Metode yang digunakan adalah *Software Testing Life Cycle (STLC)*. *STLC* seringkali dianggap sebagai bagian dari siklus hidup pengembangan perangkat lunak(Ruliansyaha et al., 2023) dan merupakan proses terstruktur yang digunakan dalam pengujian perangkat lunak. Proses yang terlibat dalam *STLC* terdiri dari setiap tahap yang memiliki kriteria dan hasil yang ditetapkan. Dalam proses *STLC*, terdapat beberapa kegiatan yang dilakukan untuk meningkatkan kualitas produk. *STLC* mengacu pada tahap-tahap yang spesifik dalam proses pengujian untuk memastikan kualitas produk. Terdapat beberapa tahap dalam *STLC* diantaranya adalah sebagai berikut :



Gambar 1. Tahapan STLC

a. *Requirement Analysis*

Pada software testing life cycle (STLC) tahapan yang paling pertama adalah requirement analysis. Pada tahapan ini dimulai dengan memahami kebutuhan perangkat lunak yang akan diuji. Merencanakan apa yang akan di uji, memikirkan kemungkinan munculnya bug disetiap halaman yang diuji dan output diharapkan oleh penguji.

b. *Test Planning*

Pada tahap test planning, Software quality assurance (SQA) mengumpulkan semua persyaratan yang diperlukan seperti memahami dokumen spesifikasi untuk mengidentifikasi kebutuhan pengguna dan mempersiapkan rencana pengujian[18]. Hasil dari proses test planning merupakan sebuah dokumen yang berisi perencanaan pengujian yang akan digunakan dalam test case. Kemudian, SQA akan mulai mengembangkan test case (Ruliansyaha et al., 2023).

c. *Test Case Development*

Pada tahap ini merupakan tahap dimana terdapat beberapa proses terjadi diantaranya adalah pembuatan, verifikasi dan pengerjaan butir uji serta test script yang telah dipersiapkan pada test planning. Awalan dari tahap ini adalah mengidentifikasi data uji serta ditinjau hingga kemudian dikerjakan ulang sesuai dengan persyaratan yang telah dipenuhi (Arfan & Hendrik, 2022).

d. *Test Environment Setup*

Proses yang dilakukan pada tahap ini adalah memastikan environment test dapat berjalan dengan baik dan sesuai dengan harapan. Pada dasarnya environment test menentukan syarat software yang diujikan mulai melakukan smoke test untuk memastikan environment test apakah sudah benar-benar siap. Pengertian dari Smoke test sendiri adalah pengujian yang dilakukan untuk memastikan fitur atau fungsi penting dari suatu program berjalan dengan baik sebelum dilakukannya tes fungsional atau regression (Dhaifullah et al., 2022).

e. *Test Execution*

Tahap selanjutnya yaitu test execution, setelah software siap uji dan persiapan pengujian siap. Pada tahap ini pengujian dilakukan berdasarkan test plan dan test case yang disepakati di tahap sebelumnya. Fitur yang berjalan sesuai dengan test requirement, status fitur tersebut adalah pass atau berhasil, dan siap untuk masuk ke tahap deployment. Jika fungsi tidak berjalan sesuai dengan requirement, maka tergantung dari kategori error atau bug yang terjadi. Error yang ditemukan tersebut dimasukkan ke dalam test report tim QA untuk disampaikan ke tim developer untuk diperbaiki, dan kembali diuji oleh tim QA. Pada tahap ini juga dilakukan sanity testing dan regression testing ketika akan melakukan deployment program (Dhaifullah et al., 2022).



f. Test Cycle Closure

Semua aktivitas terkait pengujian pada suatu perulangan diselesaikan pada tahap akhir dari Life Cycle of Software Testing. Pada tahap ini, hasil pengujian didokumentasikan, dan pelajaran dari pengujian saat ini digunakan untuk meningkatkan proses pengujian di masa depan (Yasmine Shalsabilla et al., 2024).

3. ANALISA DAN PEMBAHASAN

3.1 Kasus Uji dan Skenario

Skenario pengujian adalah suatu proses yang digunakan untuk mendapatkan data yang diinginkan dari sebuah penelitian. Metode yang digunakan untuk membuat skenario pengujian fungsionalitas adalah metode *black box*. Metode ini berfokus pada nilai *output* yang dihasilkan dari input yang diberikan, serta proses yang terjadi di dalam sistem. Skenario pengujian aplikasi *Access by KAI* terdapat di tabel berikut.

Table 1. Kasus Uji Skenario

Kelas Uji	Detail Uji	Jenis Pengujian
Halaman Login	Uji Detail Login	Black Box
Halaman Akun	Uji Detail Akun	Black Box
Halaman Kereta Antar Kota	Uji Detail Kereta Antar Kota	Black Box
Halaman Kereta Lokal	Uji Detail Kereta Lokal	Black Box
Halaman Kereta Commuter Line	Uji Detail Kereta Commuter Line	Black Box
Halaman Kereta Woosh	Uji Detail Woosh	Black Box

4. IMPLEMENTASI

Hasil pengujian *alfa* menggunakan metode *black box* dilakukan untuk memastikan apakah aplikasi dapat berjalan dengan baik (Masripah & Ramayanti, 2020). Berdasarkan hasil pengujian *alfa* melalui uji *test case*, menunjukkan beberapa *insight* sebagai berikut.

a. Hasil Uji Coba Menu Login

Table 2. Test Case Menu Login

Kode Uji	Deskripsi Uji Kasus	Hasil yang Diharapkan	Hasil Aktual	Hasil Uji
A.01	User Mendaftar Menggunakan Akun Goegle Dengan Menekan Tombol Sign In Pada Halaman Log In	User berhasil masuk ke halaman dashboard	User berhasil masuk ke halaman dashboard	[✓] Berhasil [] Gagal
A.02	User Mendaftar Menggunakan Nomor Telepon Dengan Menekan Register Pada Halaman Log In	User mengisi nomor telepon yang akan didaftarkan	User berhasil masuk ke halaman dashboard	[✓] Berhasil [] Gagal

b. Hasil Uji Coba Menu Data Diri

Table 3. Test Case Menu Data Diri

Kode Uji	Deskripsi Uji Kasus	Hasil yang Diharapkan	Hasil Aktual	Hasil Uji
B.01	User Mengisi Data diri	User berhasil menjadi member Acces KAI	User berhasil menjadi member Acces KAI	[✓] Berhasil [] Gagal
B.02	User mengisi No id (KTP) tidak sesuai dengan ketentuan yang seharusnya 16	User gagal menjadi member Acces KAI dan menerima peringatan No Id harus sebanyak 16 digit	User berhasil menjadi member Acces KAI	[] Berhasil [✓] Gagal

c. Hasil Uji Menu Kereta Antar Kota

Table 4. Test Case Menu Kereta Antar Kota

Kode Uji	Deskripsi Uji Kasus	Hasil yang Diharapkan	Hasil Aktual	Hasil Uji
C.01	User memilih stasiun asal dan stasiun tujuan	Menampilkan tiket dari berbagai tanggal dan kelas yang berbeda	Menampilkan tiket dari berbagai tanggal dan kelas yang berbeda	[✓] Berhasil [] Gagal
C.02	User memilih stasiun asal dan stasiun tujuan yang sama	Tidak ada tiket yang di tampilkan	Tidak ada tiket yang di tampilkan	[✓] Berhasil [] Gagal

d. Hasil Uji Menu Kereta Lokal

Table 5. Tes Case Menu Kereta Lokal

Kode Uji	Deskripsi Uji Kasus	Hasil yang Diharapkan	Hasil Aktual	Hasil Uji
D.01	User hanya memilih stasiun asal	User menerima alert untuk mengisi stasiun tujuan terlebih dahulu	User menerima alert untuk mengisi stasiun tujuan terlebih dahulu	[✓] Berhasil [] Gagal
D.02	User mengosongkan	User menerima alert untuk mengisi stasiun	User menerima alert untuk mengisi stasiun	[✓] Berhasil



	stasiun asal dan stasiun tujuan	asal dan tujuan terlebih dahulu	asal dan tujuan terlebih dahulu	 [] Gagal
--	---------------------------------	---------------------------------	---------------------------------	------------------

e. Hasil Uji Menu Kereta Computer Line

Table 6. Test Case Menu Kereta Commuterline

Kode Uji	Deskripsi Uji Kasus	Hasil yang Diharapkan	Hasil Aktual	Hasil Uji
E. 01	User menekan cek posisi kereta	User dapat masuk ke halaman Posisi Kereta	User dapat masuk ke halaman Posisi Kereta	[✓] Berhasil [] Gagal
E.02	User menekan menu lihat jadwal kereta	Menampilkan jadwal kereta sesuai dengan stasiun asal dan stasiun tujuan yang telah	Menampilkan jadwal kereta sesuai dengan stasiun asal dan stasiun tujuan yang telah	[✓] Berhasil [] Gagal
E.03	User menekan menu rute dan jalur	Menampilkan rute commuter line di jabodetabek dan yogyakarta	Menampilkan rute commuter line di jabodetabek dan yogyakarta	[✓] Berhasil [] Gagal

f. Hasil Uji Menu Kereta Whoosh

Table 7. Test Case Menu Kereta Whoosh

Kode Uji	Deskripsi Uji Kasus	Hasil yang Diharapkan	Hasil Aktual	Hasil Uji
F.01	Pengisian data penumpang	User tidak dapat menyimpan data penumpang karena format data tidak sesuai	User tidak dapat melakukan penyimpanan data	[✓] Berhasil [] Gagal
F.02	Jadwal whoosh dan KA Feeder	User dapat melihat jadwal dan peta whoosh dan KA Feeder	User dapat melihat jadwal dan peta whoosh dan KA Feeder	[✓] Berhasil [] Gagal
F.03	Ketentuan boarding	User dapat melihat ketentuan boarding	User dapat melihat ketentuan boarding	[✓] Berhasil [] Gagal



Hasil pengujian *alfa* mengungkapkan bahwa bug terjadi pada menu data diri. Dari skenario pengujian yang melibatkan pengisian data diri dan simulasi proses pemesanan diketahui bahwa aplikasi memungkinkan pengguna dapat menyimpan NIK kurang atau lebih dari 16 digit tanpa memberikan peringatan. *Bug* ini menyebabkan masalah ketika pengguna melakukan pemesanan, karena aplikasi menolak pemesanan tersebut dengan pemberitahuan bahwa NIK tidak sesuai format. Seharusnya, aplikasi memberikan peringatan mengenai format NIK yang tidak *valid* saat pengguna mengisi data diri, sehingga pengguna dapat segera memperbaiki kesalahan tersebut.

5. KESIMPULAN

Pengujian ini menghasilkan kesimpulan bahwa aplikasi *Access by KAI* memerlukan beberapa perbaikan pada proses pengembangan selanjutnya. Selama pengujian *alfa testing*, penulis menemukan *bug* pada menu data diri. *Bug* ini perlu diperbaiki agar seluruh fungsionalitas aplikasi *Access by KAI* dapat berjalan dengan baik dan memenuhi kebutuhan pengguna.

REFERENCES

- Amalia, A., Putri Hamidah, S. W., & Kristanto, T. (2021). Pengujian Black Box Menggunakan Teknik Equivalence Partitions Pada Aplikasi E-Learning Berbasis Web. *Building of Informatics, Technology and Science (BITS)*, 3(3), 269–274. <https://doi.org/10.47065/bits.v3i3.1062>
- Arfan, A., & Hendrik. (2022). Penerapan STLC dalam Pengujian Automation Aplikasi Mobile (Studi kasus: LMS Amikom Center PT.GIT Solution). *Jurusan Informatika, Universitas Islam Indonesia*, 3(2), 1–6. <https://jurnal.uui.ac.id/AUTOMATA/article/view/24127/>
- Dhaifullah, I. R., Muttanifudin H, M., Ananda Salsabila, A., & Ainul Yaqin, M. (2022). Survei Teknik Pengujian Software. *Journal Automation Computer Information System*, 2(1), 31–38. <https://doi.org/10.47134/jacis.v2i1.42>
- Haryanto, A., Naunsaadjie, M. A., Latief, M., & Maulana, I. (2023). Pengujian Black Box Pada Pada Sistem Informasi Hewan Qurban Berbasis Website Menggunakan Metode Teknik Equivalence Partitions. *OKTAL : Jurnal Ilmu Komputer Dan Science*, 2(6), 1621–1624. <https://journal.mediapublikasi.id/index.php/oktal>
- Ismail, Rahman Fadhlir, Dimiyati Imtiyas, Kholbi S Rimba, & Sarifudin Aries. (2023). Pengujian black box pada aplikasi pengajuan cuti karyawan griya yatim dan dhuafa menggunakan teknik equivalence partitions. *Nautical : Jurnal Ilmiah Multidisiplin*, 1(10), 1228–1234.
- Masripah, S., & Ramayanti, L. (2020). Penerapan Pengujian Alpha Dan Beta Pada Aplikasi Penerimaan Siswa Baru. *Swabumi*, 8(1), 100–105. <https://doi.org/10.31294/swabumi.v8i1.7448>
- Muhammad Arofiq, N., Ferdo Erlangga, R., Irawan, A., & Saifudin, A. (2023). OKTAL : Jurnal Ilmu Komputer dan Science Pengujian Fungsional Aplikasi Inventory Barang Kedatangan Dengan Metode Black Box Testing Bagi Pemula. *Ilmu Komputer Dan Science*, 2(5), 1322–1330. <https://journal.mediapublikasi.id/index.php/oktal>
- Putri, D. I. (2022). Teknik Equivalence Partitions untuk Pengujian Aplikasi Manajemen Kas dan Inventaris Berbasis Web. *INFORMATION MANAGEMENT FOR EDUCATORS AND PROFESSIONALS : Journal of Information Management*, 6(2), 193. <https://doi.org/10.51211/imbi.v6i2.1922>
- Ruliansyaha, Tukino, Hudac, B., & Hananto, A. L. (2023). Application of Software Testing Life Cycle in Automated Testing of Dzikra Platform. *CSRID Journal*, 15(1), 1–11. <https://www.doi.org/10.22303/csrj.15.1.2023.01-11%0Ahttps://csrjjournal.potensi-utama.org/index.php/CSRIDjournal/article/view/32/11>
- Yasmine Shalsabilla, S., Dyar Wahyuni, E., & Cahyo Wibowo, N. (2024). Implementasi Blackbox Automation Testing Pada Aplikasi Donor Menggunakan Framework Stlc Dalam Lingkup Pengembangan Agile Scrum. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(2), 2261–2269. <https://doi.org/10.36040/jati.v8i2.9465>