

BLACK BOX TESTING EQUIVALENCE PARTITIONS UNTUK PENGUJIAN FRONT-END PADA SISTEM AKADEMIK SITODA

Dahlia Widhyaestoeti¹, Saidul Iqram², Siti Nur Mutiyah³, Yasmin Khairunnisa⁴
Program Studi Teknik Informatika, Fakultas Teknik & Sains²

Universitas Ibn Khaldun Bogor

Jl.KH Sholeh Iskandar Km 2 Kota Bogor

dahlia@uika-bogor.ac.id¹, saiduliqram1@gmail.com², sitinurmutiyah31@gmail.com³,
yasminkhairunnisa1@gmail.com⁴

Abstrak

Aplikasi antar muka pengguna (*user interface*) atau disebut juga *front-end* Sistem akademik SITODA yang menarik dan dapat mengatasi permasalahan data-data akademik sangat dibutuhkan oleh bagian kurikulum dan guru pada sekolah SMKIT Insan Toda. Pengujian sistem diperlukan sebelum sistem tersebut diterapkan. *Front-end* Sistem Akademik SITODA Berbasis Web dibuat menggunakan framework CI dan bahasa pemrograman CSS. Pengujian *front-end* sistem akademik SITODA menggunakan *Black Box*, pengujinya bisa bukan orang teknis dari bidang pemrograman, maka untuk pengujian sistem melibatkan seorang bagian kurikulum yang menguji halaman admin dan guru yang akan menguji halaman guru. Pengujian *equivalence partitioning* untuk sistem akademik SITODA dirancang dengan memeriksa keluaran dan masukan data, dimulai dengan membuat tabel perencanaan pengujian yang berisi kelas uji dan butir uji, selanjutnya dilakukan pengujian untuk setiap butir uji dengan membuat tabel pengujian fungsional. Hasil pengujian *Equivalence Partition* (EP) bahwa sistem akademik SITODA berjalan sesuai dengan fungsinya karena semua hasil uji berhasil dijalankan sesuai hasil yang diharapkan.

Kata kunci :

Front-end, Black Box, Equivalence Partition, Kelas Uji

Abstract

User interface or also called front-end Academic system SITODA which interesting and can overcome the problem of academic data is needed by the curriculum and teachers in schools of SMKIT Insan Toda. System testing is required before the system is implemented. Front-end of Academic system SITODA

Web-Based created using CI Framework and CSS Programming language. The front-end testing of the SITODA academic system uses Black Box, the examiner may not be a technical person from the programming field, system testing involves a curriculum section testing the admin form and a teacher who will test the teacher form. The equivalence partitioning test for the SITODA academic system is designed by checking the output and input data, starting with making a test planning table containing test classes and test items, then testing for each test item by creating a functional test table. The results of the Equivalence Partition (EP) test show that the SITODA academic system is running according to its function because all test results have been successfully executed according to the expected results.

Keywords :

Front-end, Black Box, Equivalence Partition, Test Class.

I. PENDAHULUAN

SMKIT Insan Toda merupakan salah satu sekolah swasta yang berada di Kota Bogor, proses pengolahan data akademik di sekolah ini dari Sistem akademik yang sedang berjalan masih menggunakan aplikasi *Microsoft. Excel*. Pengolahan dan organisasi data yang masih menggunakan buku daftar kelas tentu merupakan tantangan tersendiri bagi guru dan bagian kurikulum pada saat kebutuhan data digunakan untuk mengolah hasil akademik para siswa.

Untuk meningkatkan efektifitas dan efisiensi pengelolaan akademik diperlukan sebuah Sistem Informasi Akademik (SIA) yang diterapkan di sekolah

tersebut (Purwanto, 2017). Kehilangan data, kebutuhan waktu dalam penulisan buku daftar kelas dan proses perhitungan serta penilaian akademik siswa adalah permasalahan yang dialami oleh guru di SMKIT Insan Toda. Dengan permasalahan tersebut perlu dibangun sebuah sistem akademik berbasis website dengan tampilan yang menarik yang dapat menampilkan informasi data secara akurat dan terkini, serta dapat melihat data nilai yang telah dilakukan oleh bidang kurikulum.

Front-end adalah sebuah bagian dari sistem yang menyediakan tampilan kepada pengguna. Bertugas untuk mengembangkan komponen-komponen visual pada suatu sistem dan bertanggung jawab pada tampilan antarmuka (Rizaldy & Dirgahayu, 2020). Menurut (Aslamiyah & Kusumas Sita, 2019) *Front-End* merupakan istilah yang mengacu pada developer yang bekerja dibagian desain layout suatu website atau aplikasi, dimana seorang developer akan mengerjakan user interface agar desain terlihat menarik dan tidak terlihat membosankan untuk dilihat. Seorang *Front-End* developer wajib menguasai program HTML, CSS, Javascript dan juga framework seperti jQuery, Bootstrap dan lain sebagainya.

Front-end Sistem Akademik SITODA Berbasis Web menggunakan framework CI dan bahasa pemrograman CSS untuk tampilan login, tampilan siswa, tampilan guru dan tampilan admin kurikulum. Setelah Sistem akademik SITODA dibuat diperlukan pengujian sistem dengan *black box*, pengujian ini dilakukan langsung pada pengguna yaitu bagian kurikulum dan guru, agar developer mengetahui kesalahan dan kekurangan fungsional dari sistem ini.

II. KAJIAN LITERATUR

II.1 User Interface

Sebuah aplikasi perangkat lunak dibentuk dari dua macam aplikasi, yaitu aplikasi basis data dan aplikasi antar muka pengguna (*user interface*). Aplikasi basis data disebut *back-end*, sedangkan aplikasi *user interface* disebut *front-end*. (Subroto, 2020, p.4).

Menurut Lastiansah (2012) dikutip dalam penelitian (R et al., 2020), *User interface* adalah cara program dan pengguna untuk berinteraksi. *User interface* yang dibuat pada sistem akademik SITODA digunakan oleh siswa, guru dan bagian kurikulum.

Sistem Informasi Akademik dikembangkan sebagai layanan informasi berupa data-data yang berhubungan dengan data akademik (Purwanto, 2017). Sistem Informasi Akademik SITODA dibuat untuk mengelola data siswa, data guru serta pelaporan hasil akademik masih dilakukan secara sederhana dimana data hanya disimpan dalam aplikasi penyimpanan data *Microsoft excel*.

II.2 Pengujian Perangkat Lunak

Menurut K. Sneha and G. M. Malle (2017) dalam (Arfinda et al., 2019) untuk mengantisipasi dampak kerugian kedepan yang ditimbulkan, maka dilakukan suatu proses yaitu pengujian. Bentuk pengujian yang digunakan dapat seperti *white box testing* atau *glass box testing*, *black box testing (behavioral testing)* dan *grey box testing*. *White box testing* dilakukan pengujian berdasarkan pengecekan secara detail pada perancangan program, struktur kontrol desain program secara prosedural untuk membagi pengujian ke dalam beberapa kasus pengujian. *Black box testing* dilakukan pengujian dengan mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari aplikasi yang diuji.

Khusus pada web, pengujian berarti suatu teknologi web yang menerima pesan dari pengguna dan memberikan respon yang sesuai melalui protokol atau aturan yang ada. Pengujian *black box* memiliki peran penting dalam pengujian perangkat lunak yaitu untuk memvalidasi fungsi keseluruhan sistem apakah telah bekerja dengan baik, serta pengujian *black box* bersifat dinamis. Pengujian *black box* harus membuat kasus uji dengan dua perbandingan antara benar atau salah (Arfinda et al., 2019).

Dalam studi literatur yang dilakukan Parlika dkk, ada 8 tujuan dari *Black Box Testing* adalah untuk mencari fungsi yang salah atau hilang, kesalahan antar muka, kesalahan dalam struktur data atau akses database eksternal, kesalahan kinerja, inisialisasi dan kesalahan terminasi, validasi fungsional, kesensitifan sistem terhadap nilai input tertentu, dan batasan suatu data (Parlika et al., 2020).

Parlika dkk juga menuliskan ada 8 metode pengujian *Black Box: Equivalence Partitioning, Boundary Value Analysis, Cause Effect Graph, Random Data Selection, Feature Test, All- Pair Testing, Fuzzing, Orthogonal Array Testing*. Hasil dari pengelompokan terhadap seluruh literatur, menyebutkan 9 kekurangan dan 10 kelebihan literatur pengujian *Black Box*. Tidak semua produk perangkat

lunak dapat diuji adalah salah satu kekurangan dari pengujian *Black Box*. *Tester* tidak perlu tahu bahasa pemrograman atau bagaimana pelaksanaan program adalah satu kelebihan dari pengujian *Black Box*, maka di simpulkan bahwa pengujian *Black Box* bisa bukan orang teknis dari bidang pemrograman (Parlika et al., 2020).

Menurut (Mustaqbal et al., 2015) langkah-langkah pengujian software ada 4 yaitu:

1. *Unit testing-testing* per unit yaitu mencoba alur yang spesifik pada struktur modul kontrol untuk memastikan pelengkapan secara penuh dan pendeteksian error secara maksimum
2. *Integration testing* – testing per penggabungan unit yaitu pengalamatan dari isu-isu yang diasosiasikan dengan masalah ganda pada verifikasi dan konstruksi program
3. *High-order test* yaitu terjadi ketika software telah selesai diintegrasikan atau dibangun menjadi satu –tidak terpisah-pisah
4. *Validation test* yaitu menyediakan jaminan akhir bahwa software memenuhi semua kebutuhan fungsional, kepribadian dan performa.

II.3 Black Box Testing - Equivalence Partition

Equivalence Partitioning adalah sebuah metode pengujian berdasarkan masukan data pada setiap form yang ada pada sistem aplikasi informasi data kinerja, dimana setiap menu masukan akan dilakukan pengujian dan juga dikelompokkan berdasarkan fungsinya baik itu bernilai valid maupun tidak valid. Pengujian *equivalence partitioning* untuk perangkat lunak dapat dirancang dengan memeriksa keluaran dan masukan data (Parlika et al., 2020).

Pengujian fungsional untuk Sistem Pakar Diagnosa Penyakit Pada Ikan Budidaya Air Tawar dilakukan dengan melakukan pengujian fungsional menggunakan metode *black box equivalence partitioning* (EP) untuk menemukan kesalahan-kesalahan pada sistem yang telah dibangun. Pengujian dilakukan dengan membagi domain masukan (input) ke dalam kelas-kelas sehingga *test case* pada aplikasi dapat diperoleh. Pengujian dilakukan untuk menguji fungsi-fungsi sistem terutama menguji aturan (menambah, mengedit, dan menghapus) dan menguji hasil diagnosa sistem pakar berdasarkan fakta-fakta yang diberikan. (Aristoteles et al., 2015).

Teknik *equivalence partitioning* juga digunakan untuk menguji Sistem informasi akademik MTS Al-Hidayah dengan membuat tabel rancangan *Test Case* yang berfungsi untuk menyimpulkan apakah sistem berhasil dalam pengujian tipe tersebut atau tidak (Priyaungga et al., 2020).

Pada pengujian *Black Box* untuk Aplikasi Sistem Kasir Berbasis Website dengan metode *equivalence partitioning*, kesimpulan dibuat berdasarkan penilaian kesesuaian antara hasil pengujian dengan hasil yang diharapkan. Jika hasil pengujiannya sesuai harapan maka disimpulkan sesuai/berhasil. Jika hasil pengujiannya sesuai hasil yang diharapkan maka disimpulkan salah/gagal (Aziz et al., 2020).

III. PENGUJIAN BLACK BOX TESTING TEKNIK EQUIVALENCE PARTITIONS

Membuat tabel pengujian yang berisi kelas uji dan butir uji dilakukan dalam melakukan pengujian *Black Box* untuk menguji apakah program berjalan sesuai spesifikasi kebutuhan (requirement) (Kholipa et al., 2021).

Pengujian *Equivalence Partition* (EP) dimulai dengan membuat tabel perencanaan pengujian yang berisi kelas uji dan butir uji. Tabel 1 berisi kolom kelas uji dan butir uji untuk pengujian sistem akademik SITODA.

Tabel 1. Rencana Kelas Uji dan Butir Uji

No	Kelas Uji	Butir Uji
1	Login	Menu Login
2	Halaman Admin	Menu Data Guru, Menu Data Siswa, Menu Data Mapel, Menu Data Kelas, Menu Data Ekstra, Menu Set Data Kelas, Menu Set Data Wali Kelas, Menu Set Tahun Aktif, Menu Set Data Mapel.
3	Halaman Guru	Menu Mapel Diampu, Menu Penilaian Siswa.

Selanjutnya dilakukan pengujian untuk setiap butir uji dengan membuat tabel pengujian fungsional yang berisi kolom-kolom: kelas uji, pengujian keterbatasan, skenario uji, hasil yang diharapkan dan hasil uji. Pengujian EP pada tahap ini melibatkan seorang Bagian Kurikulum yang menguji halaman admin dan guru yang akan menguji halaman guru. Tabel 2, berisi hasil pengujian fungsionalitas sistem akademik SITODA.

Tabel 2. Hasil Pengujian Equivalence Partition (EP)

Form Uji	Uji Keterbatasan	Skenario Uji	Hasil Yang diharapkan	Hasil Uji
Menu Login	User Admin, User Guru dan Siswa yang sudah di aktifkan akunnya	Masukkan data Username dan password yang benar	Ketika masuk ke dalam menu login akan muncul form login dan ketika form login di isi dengan username dan password yang benar maka akan muncul notifikasi berhasil login lalu akan ke menu Dashboard.	[√] Berhasil [] Gagal
			Ketika masuk ke dalam menu login akan muncul form login dan ketika form login di isi dengan username dan password yang salah maka akan muncul notifikasi gagal login lalu akan kembali ke menu login.	[√] Berhasil [] Gagal
			Ketika menu data guru diklik, maka akan tampil tabel data guru.	[√] Berhasil [] Gagal
Mengaktifkan Akun User Guru	Hanya admin yang dapat melihat data guru	View menu data guru	Ketika data dipilih dan kemudian klik tombol aktifkan akun, maka akan muncul notifikasi akun guru berhasil di aktifkan dan user guru yang dipilih dapat digunakan untuk mengakses sistem SITODA.	[√] Berhasil [] Gagal
			Ketika data dipilih dan kemudian klik	[√] Berhasil [] Gagal
			Ketika data dipilih dan kemudian klik	[√] Berhasil [] Gagal

Form Uji	Uji Keterbatasan	Skenario Uji	Hasil Yang diharapkan	Hasil Uji
		n akun guru	tombol non-aktifkan akun, maka akan muncul notifikasi akun guru berhasil di non-aktifkan dan user guru yang dipilih tidak dapat mengakses sistem SITODA.	
Menu Data Siswa	Hanya admin dan siswa yang dapat melihat data nilai siswa, dan mencetak rapor	View Data Siswa	Ketika menu siswa diklik, akan tampil tabel dan data Siswa, dan akan menampilkan rapor siswa pada sistem akademik SITODA	[√] Berhasil [] Gagal
			Ketika data nilai pada form diisi lengkap dan diklik tombol tambah, maka data nilai akan ditambahkan dan tersimpan di database.	[√] Berhasil [] Gagal
			Ketika data nilai pada form edit dan klik tombol edit, maka data akan teredit dan tersimpan di database.	[√] Berhasil [] Gagal
Manajemen Data Nilai	Hanya guru yang dapat melihat tampilan data nilai siswa	View data Nilai	Ketika Menu Data Nilai di klik maka akan muncul Tabel Nilai	[√] Berhasil [] Gagal
			Ketika data pada form diisi lengkap dan diklik tombol tambah, maka data nilai akan ditambahkan dan tersimpan di database.	[√] Berhasil [] Gagal

Form Uji	Uji Keterbatasan	Skenario Uji	Hasil Yang diharapkan	Hasil Uji
	Hanya guru yang dapat mengedit nilai siswa	Form edit untuk mengedit data	Ketika data diedit pada form edit dan klik tombol edit, maka data akan teredit dan tersimpan di database.	[√] Berhasil [] Gagal

Hasil dari pengujian *Equivalence Partition* (EP) pada sistem akademik SITODA tidak ada yang gagal untuk setiap butir uji. Dapat disimpulkan dari hasil pengujian *Equivalence Partition* (EP) bahwa sistem akademik SITODA berjalan sesuai dengan fungsinya dalam mengelola aturan tambah dan edit karena tidak memiliki kegagalan.

IV. KESIMPULAN DAN SARAN

Kesimpulan dari pengujian *black box testing* dengan teknik *Equivalence Partition* pada sistem akademik SITODA adalah:

1. Pembuatan *front-end* pada Sistem Akademik SITODA Berbasis Web (Studi kasus: SMKIT Insan Toda) telah dilakukan dan diimplementasikan berdasarkan hasil perancangan yang telah dibuat. *Front-end* Sistem Akademik SITODA Berbasis Web dibuat menggunakan framework CI dan bahasa pemrograman CSS untuk tampilan login, tampilan siswa, tampilan guru dan tampilan admin kurikulum telah berhasil melalui tahap pengujian serta saran dari pihak sekolah untuk mendapatkan informasi perkembangan hasil sekolah.
2. Sistem ini yang telah dibuat dapat mempermudah murid dan guru untuk melihat data-data akademik berupa data nilai dan biodata guru berjalan dengan baik.

Saran dari pengujian *Equivalence Partition* pada sistem akademik SITODA adalah:

1. Penambahan beberapa fitur yang di sarankan oleh pihak sekolah
2. Pengujian sistem akademik SITODA dilakukan dengan metode lain dan jumlah responden yang dapat menentukan nilai fungsional dan kesalahan dari sistem tersebut.

REFERENSI

- Subroto, B. *Back-End SQL Server Express dan Front-End Access: Membuat Aplikasi Dokumen Elektronik Sederhana*, p.4. 5 Agustus, 2020, from <https://www.google.co.id/books/edition/Back-End-SQL-Server-Express-dan-Front-End-Czf1DwAAQBAJ?hl=id&gbpv=1&kptab=overview>
- Arfinda, D., Selo, S., & Lukito, L. (2019). *Studi Literatur Pengujian Perangkat Lunak*. 15–21. <https://unisbank.ac.id/ojs/index.php/sintak/article/view/7560>
- Aristoteles, Wardiyanto, & Pratama, A. A. (2015). Sistem Pakar Diagnosa Penyakit Pada Ikan Budidaya Air Tawar dengan Metode Forward Chaining. *Jurnal Komputasi*, 3(2), 99–108.
- Aslamiyah, S., & Kusumas Sita, Y. (2019). Membangun Aplikasi Front-End Pelayanan Petugas Bank Sampah Building A Front-End Application For Waste Bank Service Officers. *SNPMas : Seminar Nasional Pengabdian Pada Masyarakat*, 111–118. <https://ejournal.diponegara.ac.id/index.php/snpmas/article/view/435>
- Aziz, I. A., Setiawan, B., Khanh, R., Nurdiansyah, G., & Yulianti, Y. (2020). Pengujian Black Box pada Aplikasi Sistem Kasir Berbasis Website Menggunakan Teknik Equivalence Partitions. *Jurnal Teknologi Sistem Informasi Dan Aplikasi*, 3(2), 82. <https://doi.org/10.32493/jtsi.v3i2.4693>
- Kholipa, Y. N., Widyaestoeti, D., Kom, S., Kom, M., Eko, A., Pramuko, K., Si, S., Studi, P., Informatika, T., & Teknik, F. (2021). *PENERAPAN NEAR FIELD COMMUNICATION PADA SISTEM PEMBAYARAN OTOMATIS TRANSPORTASI BUS BERBASIS INTERNET OF THINGS*. 7(2), 184–192.
- Mustaqbal, M. S., Firdaus, R. F., & Rahmadi, H. (2015). *PENGUJIAN APLIKASI MENGGUNAKAN BLACK BOX TESTING BOUNDARY VALUE ANALYSIS (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN)*. 1(3), 31–36.
- Parlika, R., Nisaa, T. A., Ningrum, S. M., & Haque, B. A. (2020). Studi Literatur Kekurangan dan

Kelebihan Pengujian Black Box. *Teknomatika*, 10(02), 131–140.

- Priyaungga, B. A., Aji, D. B., Syahroni, M., Aji, N. T. S., & Saifudin, A. (2020). Pengujian Black Box pada Aplikasi Perpustakaan Menggunakan Teknik Equivalence Partitions. *Jurnal Teknologi Sistem Informasi Dan Aplikasi*, 3(3), 150. <https://doi.org/10.32493/jtsi.v3i3.5343>
- Purwanto, R. (2017). Peningkatan Efektifitas Dan Efisiensi Pengelolaan. *Jurnal Teknologi Terapan*, 3(2), 24–31.
- R, F. F., Riko, N., Zaidiah, A., & Kom, S. (2020). *Perancangan Desain User Interface Lost And Found*. 2012, 312–318. <https://conference.upnvj.ac.id/index.php/senami/ka/article/view/349/378>
- Rizaldy, R., & Dirgahayu, R. T. (2020). Pengembangan Front-End Sistem Informasi Pendataan Pendar Foundation Yogyakarta. *Automata*, 1(2), 190–194. <https://journal.uii.ac.id/AUTOMATA/article/view/15591/10252>