

Otomatisasi Pengujian Sistem Informasi Koperasi Berbasis Web Menggunakan Teknik *Equivalence Partitioning*

Ivana Yunitasari^{1*}, Muhammad Deva Ditgara¹, Anggi Febriana¹, Yeni Rahmawati¹,
Ahmad Fauzi¹

¹Fakultas Ilmu Komputer, Teknik Informatika, Universitas Pamulang, Jl. Raya Puspipetek No. 46,
Kel. Buaran, Kec. Serpong, Kota Tangerang Selatan. Banten 15310, Indonesia

Email: ^{1*}yunitasariivana@gmail.com, ²devagr@gmail.com, ³anggi.febriana@gmail.com,
⁴yenir1130@gmail.com, ⁵dosen02621@unpam.ac.id

(* : coressponding author)

Abstrak—Proses validasi yang dilakukan untuk mengetahui kualitas dari suatu perangkat lunak merupakan pengertian dari pengujian perangkat lunak. Pengujian ditujukan untuk mengetahui kesalahan keluaran aplikasi yang mengakibatkan sistem tidak berjalan dengan semestinya. ada 4 jenis pengujian *Black Box* di antaranya, yaitu teknik *Pairwise Testing*, *Boundary Value Analysis*, *Equivalence Partitioning* dan *Error Guessing*. memiliki sebuah kelebihan dan kekurangan sesuai dengan kebutuhan aplikasi yang diuji. dipilih lah teknik *Equivalence Partitioning* sebagai sebuah pengujian *Black box* karena sesuai dengan kebutuhan pada aplikasi Sistem Informasi Koperasi berbasis web, pada pengujian ini dibagi 2 jangkauan pengujian menjadi valid dan tidak valid. Diawali dengan mengidentifikasi, menguji fungsi sistem, menetapkan data dan arsip penelitian, Perancangan kasus percobaan serta pengujian, dan mendokumentasikan akibat penelitian dan konklusi penelitian. Hasil dari dilakukannya pengujian, bahwa perangkat lunak ini berjalan sesuai apa yang sudah diuji. Tetapi sampel pengujian dilakukan dengan beberapa jumlah formulir pembuktian, untuk sementara perangkat ini tidak ada kendala dan berjalan dengan baik.

Kata Kunci: Teknik, *Black Box*, Sistem, Aplikasi Sistem Informasi Koperasi Berbasis Web

Abstract—The validation process carried out to determine the quality of the software is an understanding of software testing. Testing is intended to find out application output errors that result in the system not running properly. There are 4 types of *Black Box* testing including *Pairwise Testing*, *Boundary Value Analysis*, *Equivalence Partitioning*, and *Error Guessing* techniques. This technique has advantages and disadvantages according to the needs of the application being tested. The *Equivalence Partitioning* technique was chosen as a *Black box* test because it is in accordance with the needs of the web-based Cooperative Information System application, in this test is divided into 2 test ranges to be valid and invalid. Beginning with identifying, testing system functions, establishing research data and archives, designing experimental cases and testing, and documenting research results and research conclusions. The results of testing, that this software runs according to what has been tested. However, the test is only carried out with a number of sample forms, for the time being this device has no problems and runs well.

Keywords: Technique, *Black Box*, System, Web-based Cooperative Information System Application

1. PENDAHULUAN

Pada saat seorang *programmer* atau para pengembang aplikasi membuat sebuah perangkat lunak, tentunya akan ditemukan kesalahan pada proses pembuatan. Untuk menghindari banyaknya *bug* atau kesalahan dari sistem yang sudah dibuat, maka perlu dilakukan pengujian perangkat lunak sebelum perangkat lunak tersebut akan di *publish* ke publik. Pengujian perangkat lunak sangatlah penting dalam menjamin kualitas program. Tujuannya yaitu memastikan apakah aplikasi sudah berjalan sebagaimana mestinya dan mendeteksi terjadi error dalam program serta pengesahan apakah sudah memenuhi keinginan pengguna atau belum. Pengujian perangkat lunak mengukur seberapa baik mutu dari suatu aplikasi, dengan cara ini aplikasi yang dibuat tidak akan crash ketika anda salah memasukkan data atau sesuatu yang tidak diinginkan.

Yang perlu diperhatikan dalam pengujian ini yaitu pengujian tersebut harus dapat mendeteksi kesalahan yang mungkin belum terdeteksi sebelumnya, jika kesalahan ini dapat diperbaiki, pengujian dikatakan berhasil maka kualitas software akan jauh lebih baik. Saat menguji hal-hal yang harus diperhatikan dalam pengujian agar dapat dengan mudah ditemukan kesalahan dan diperbaiki dengan cepat untuk menghemat waktu pengujian.

Perangkat lunak yang akan diuji yaitu sistem informasi koperasi berbasis *web* dengan tujuan untuk aplikasi keuangan dalam mengelola koperasi yang baik, sehingga dapat terstruktur dengan baik. Pengujian sistem merupakan proses validasi sebuah sistem untuk menguji apakah perangkat lunak sudah siap digunakan oleh user atau tidak. proses pengujian terdiri dari tahap-tahap dimana untuk mendapatkan desain kasus uji yang spesifik. Tujuan dari pengujian perangkat lunak adalah menemukan adanya sebuah kesalahan atau *error* dari perangkat lunak ketika mengembangkan sebuah sistem.

Galat satu jenis metode pengujian yg memperlakukan software yg tak diketahui kinerja internal adalah metode *black box*. Metode *black box* adalah metode yg simpel karena hanya diharapkan batas bawah serta batas dari sebuah data yg diharapkan, estimasi data uji dapat dicermati melalui banyaknya data entry yang akan diuji.

2. METODOLOGI PENELITIAN

2.1 *Black Box Testing*

Black-box testing adalah salah satu teknik yang digunakan untuk pengujian sebuah software atau perangkat lunak tanpa harus mengamati detail perangkat lunak. Pengujian ini hanya mengecek nilai *output* masing-masing, tidak untuk mengetahui kode program yang akan output pakai (Suherman, at all., 2019). Pengujian ini dilakukan beberapa tahapan, tahap yang pertama yaitu dengan test case pada *software* yaitu dengan menggunakan teknik *Equivalence partition* setelah itu melakukan inialisasi yaitu grade partition dari inputan dan keluaran untuk mendapatkan dokumentasi dari pengujian tersebut (Kesuma jaya, Tresnawati, Gumilang, Desyani, Andersen, 2019). Pengujian *black box* yang akan digunakan ini yaitu *teknik equivalence partitioning* yang saat ini penulis gunakan dalam pengujian inputan dan mengelompokan input sesuai fungsinya yaitu:

- Identifikasi permasalahan: Mengidentifikasi dan menganalisa supaya menemukan jalan keluar atau pemecahan masalah yang ditemukan.
- Analisa kebutuhan penguji: Menganalisa kebutuhan yang berguna untuk mempersiapkan pengujian perangkat lunak.
- Perancangan Test Case: Merancang skenario pengujian atau test case berdasarkan Analisa spesifikasi dan kebutuhan yang telah di Analisa.
- Pelaksanaan pengujian: Melaksanakan pengujian dengan menjalankan skenario pengujian yang telah dibuat.
- Hasil dan kesimpulan: Tahapan terakhir dari penelitian ini yaitu menyimpulkan hasil dari pengujian yang bisa digunakan sebagai bahan evaluasi selanjutnya.

2.2 *Teknik Equivalence Partitioning*

Teknik *Equivalence Partitions* adalah teknik uji yang menggunakan *input* untuk setiap menu yang terdapat pada sistem aplikasi presensi untuk menguji beberapa menu input dengan cara mengkategorikan dan mengelompokkan berdasarkan fungsinya (Riro, Ade, Muhammad, Daniel, & Yulianti, 2020). Metode *equivalence partitioning* yaitu terbagi menjadi kelas data dalam sistem *input domain*. Metode ini menguji *class* yang terjadi kesalahan, sehingga jumlah *test case* dapat dikurangi. Metode *equivalence partitioning* menjelaskan apakah *valid* atau tidak nya saat di *input* (Arfani, Kasih, & Pamungkas, 2021). Penentuan untuk kelas ekuivalensi sebagai berikut:

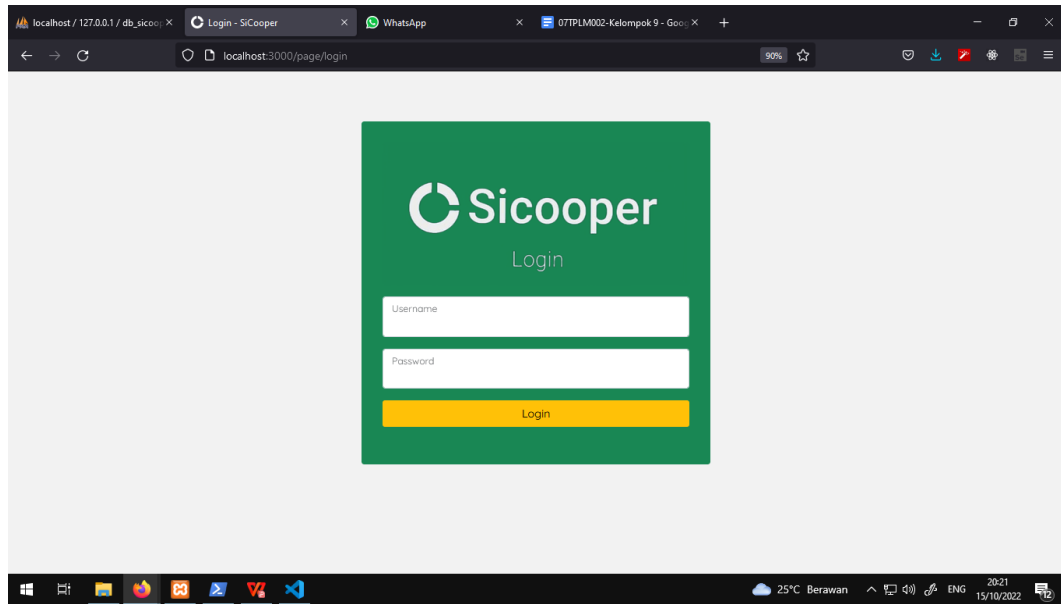
- Jika kondisi input menentukan suatu range nilai, maka akan ada satu kelas *valid* dan dua kelas yang *invalid*.
- Jika suatu kondisi menentukan anggota dalam himpunan, maka buat kelas *ekuivalensi valid* untuk semua nilai himpunan dan dua kelas *invalid*.
- Jika suatu kondisi input berupa boolean, maka akan ada kelas untuk nilai true dan false

3. ANALISA DAN PEMBAHASAN

Pengujian adalah proses uji buat menyelidiki apakah sebuah perangkat lunak mempunyai cacat atau kesalahan sistem dan memperbaikinya sehingga sistem dikatakan layak buat diberikan

kepada pengguna (Nurudin et al., 2020). Pengujian dilakukan dengan mengamati yang akan terjadi tes melalui data uji dan menyelidiki fungsi dari aplikasi. Proses pengujian menggunakan metode *Black Box* yang dimana acara yang sudah didesain dengan memasukan sebuah formulir data. Maka pengujian *Black Box* dibutuhkan guna perangkat lunak berjalan menggunakan semestinya.

Teknik *Equivalence Partitioning* menggunakan inisialisasi *standard grade partitions* diambil menjadi penelitian ini buat menguji seberapa efisiensi *software*. Hal ini dilakukan pengujian memakai metode *Equivalence Partitioning* untuk menerima data hasil pengujian dan menilai taraf efektivitas yg sudah didokumentasikan. (Kurniawan, Maulana, Sukma, Keumala, & Saifudin, 2020).

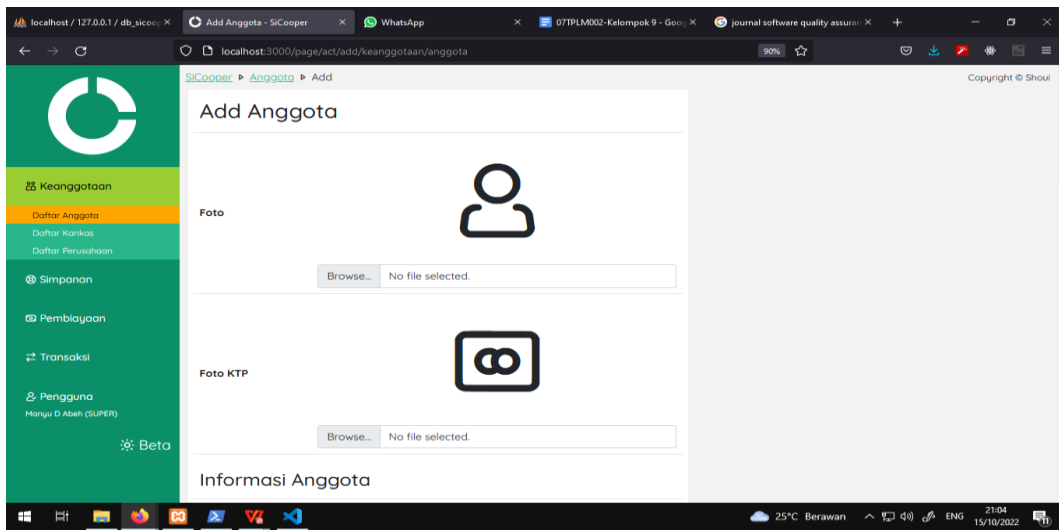


Gambar 1. Menu *Login*

Pada gambar 1 dalam menu *login* terdapat beberapa masukan dalam form login, diantaranya “*username*” dan “*password*”. pada inputan ini dilakukan uji pada kolom “*username*” akan valid jika di isi dengan *username* dan *password* yang telah terdaftar. setelah mengisi pada menu *login* dengan benar, maka akan menampilkan menu utama perangkat lunak, namun jika *username* dan *password* yang dimasukkan salah atau tidak terdaftar maka sistem akan menolak dan menampilkan “*password* atau *username* yang dimasukan salah”, begitupun jika user tidak mengisi *username* dan *password* maka sistem akan memberitahu untuk mengisi *username* dan *password* yang kosong.

Tabel 1. Uji Coba Menu *Login*

ID	Deskripsi Uji	Hasil yang Diharapkan
USR001	mengetik <i>username</i> dan <i>password</i> yang telah terdaftar username “ <i>abeh</i> ” dan <i>password</i> “ <i>abeh</i> ” lalu tekan login	Sistem menerima dan menampilkan halaman dashboard
USR002	Mengisi <i>Username</i> dan <i>Password</i> yang tidak terdaftar dengan <i>username</i> “ <i>super</i> ” dan <i>password</i> “ <i>admin</i> ”	Sistem akan menolak dan menampilkan notifikasi kesalahan
USR003	Tidak mengisi pada kolom <i>login</i>	Sistem mengarahkan pengguna agar mengisi <i>username</i> dan <i>password</i>

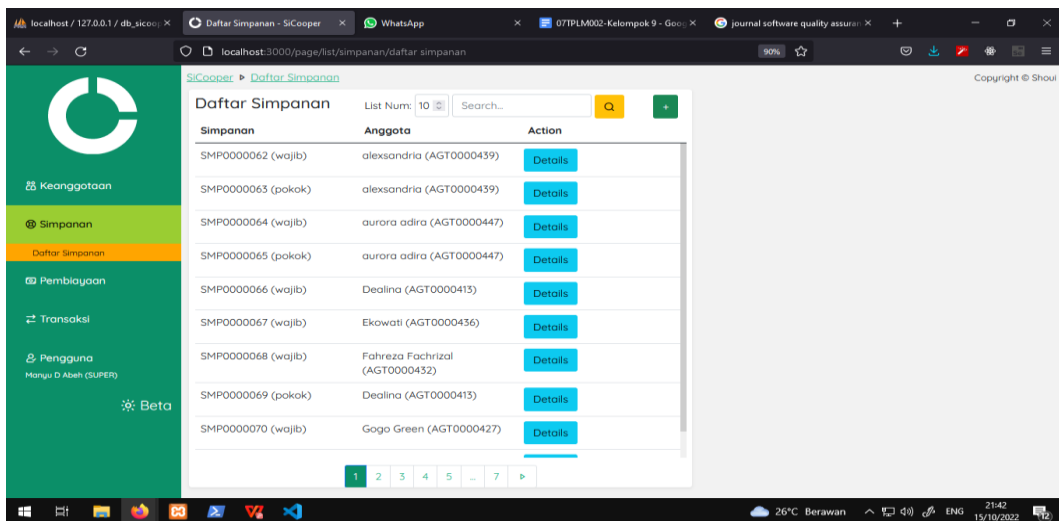


Gambar 2. Form Penambahan Anggota

Pada gambar 2 pada tahap ini ada beberapa pengujian. Dalam kolom anggota terdapat beberapa pengujian diantaranya pada kolom unggah *file* akan benar bila yang diunggah memiliki format yang sesuai dengan sistem dengan format PNG, JPG maupun JPEG, namun sistem akan menolak jika format yang dimasukkan *user* berbeda dengan format *file* yang benar.

Tabel 2. Form Penambahan Anggota

ID	Deskripsi Uji	Hasil yang Diharapkan
AGT001	Mengunggah <i>file</i> dengan format PNG/JPG/JPEG	Sistem akan menyetujui dan file akan tersimpan file akan menampilkan notifikasi “berkas berhasil diupload”
AGT002	User tidak mengunggah <i>file</i> gambar	Sistem akan mengarahkan user agar mengisi <i>file</i>
AGT003	Mengunggah file dengan format yang tidak sesuai dengan sistem	Sistem akan menolak dan menampilkan notifikasi kesalahan



Gambar 3. Menu Simpanan

Pada gambar 3 pengujian terdapat pada kolom penambahan simpanan pokok dan simpanan wajib yang akan valid apabila simpanan wajib yang dimasukan menggunakan tipe data *float* dan harus sesuai dengan id anggota terdaftar, namun jika dikosongkan sistem akan menyarankan untuk melakukan pengisian terhadap simpanan wajib dan pokok, dan jika user mengisi masukan tidak sesuai dengan tipe data yang diizinkan sistem, maka sistem akan menolak. dan apabila user mengisi simpanan pokok dan wajib tidak sesuai dengan *id* anggota maka sistem akan memberikan notifikasi kesalahan.

Tabel 3. Menu Simpanan

ID	Deskripsi pengujian	Hasil yang Diharapkan
SMP001	Menambahkan simpanan pokok dan wajib menggunakan tipe data float dan sesuai dengan id anggota yang terdaftar lalu tekan tombol submit	Sistem akan menyimpan simpanan wajib dan pokok sesuai dengan id anggota yang terdaftar
SMP002	Menambahkan simpanan pokok dan wajib menggunakan tipe data yang tidak sesuai dengan sistem lalu tekan submit	Sistem akan menolak dan tidak menampilkan masukan user
SMP003	Mengosongkan pengisian simpanan wajib dan pokok lalu tekan submit	Sistem akan menyarankan user agar mengisi form simpanan wajib dan pokok
SMP004	Menambahkan simpanan pokok dan wajib menggunakan tipe data float namun tidak sesuai dengan id anggota yang terdaftar	Sistem akan menolak dan menampilkan notifikasi salah dari sistem

4. IMPLEMENTASI

Setelah dilakukan perancangan testcase tersebut, kami melakukan beberapa dari pengujian sesuai dengan rancangan yang sudah kami buat. Jika keluaran sudah sesuai yang diharapkan user maka pengujian ini berhasil, dan jika belum sesuai maka dapat diperbaiki oleh pengembang. Hasil pengujian ini dapat dilihat pada tabel berikut.

Tabel 4. Hasil Pengujian Sistem

ID	Deskripsi Uji	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
USR001	Mengetik <i>username</i> dan <i>password</i> yang telah terdaftar username “abeh” dan password “abeh” lalu tekan login	Sistem menerima dan menampilkan halaman <i>dashboard</i>	Sistem berhasil menampilkan halaman berupa <i>dashboard</i>	Sesuai
USR002	Mengisi Username dan Password yang tidak terdaftar dengan <i>username</i> “super” dan <i>password</i> “admin”	Sistem akan menolak dan menampilkan notifikasi kesalahan	Tampil peringatan “Login gagal!, Paasword tidak sesuai”	Sesuai

USR003	Tidak mengisi pada kolom <i>login</i>	Sistem mengarahkan pengguna agar mengisi <i>username</i> dan <i>password</i>	Tampil peringatan pengarahannya pengguna untuk mengisi <i>username</i> dan <i>password</i>	Sesuai
AGT001	Mengunggah <i>file</i> dengan format PNG/JPG/JPEG	Sistem akan menyetujui dan file akan tersimpan file akan menampilkan notifikasi	Tampil sistem menyetujui file dan tersimpan pada notifikasi	Sesuai
AGT002	User tidak mengunggah <i>file</i> gambar	Sistem akan mengarahkan user agar mengisi <i>file</i>	Tampil pengarahannya user agar mengisi file dengan format sesuai	Sesuai
AGT003	Mengunggah file dengan format yang tidak sesuai dengan sistem	Sistem akan menolak dan menampilkan notifikasi kesalahan	Tampil peringatan kesalahan dan format tidak sesuai	Sesuai
SMP001	Menambahkan simpanan pokok dan wajib menggunakan tipe data float dan sesuai dengan id anggota yang terdaftar lalu tekan tombol submit	Sistem akan menyimpan simpanan wajib dan pokok sesuai dengan id anggota yang terdaftar	Sistem menyimpan simpanan wajib sesuai dengan id anggota yang terdaftar	Sesuai
SMP002	Menambahkan simpanan pokok dan wajib menggunakan tipe data yang tidak sesuai dengan sistem lalu tekan submit	Sistem akan menolak dan tidak menampilkan masukan user	Tampil sistem menolak menampilkan masukan user	Sesuai
SMP003	Mengosongkan pengisian simpanan wajib dan pokok lalu tekan submit	Sistem akan menyarankan user agar mengisi form simpanan wajib dan pokok	Tampil sistem menyarankan user agar mengisi form simpanan wajib dan pokok	Sesuai
SMP004	Menambahkan simpanan pokok dan wajib menggunakan tipe data float namun tidak sesuai dengan id anggota yang terdaftar	Sistem akan menolak dan menampilkan notifikasi salah dari sistem	Tampil sistem menolak menampilkan notifikasi salah dari sistem	Sesuai

Pada hasil akhir pengujian menggunakan *Black box* tidak ditemukan adanya error atau bug pada pengujian fungsional aplikasi melalui web. Pada bagian ini menjelaskan apakah aplikasi sudah sesuai dengan permintaan user atau belum serta kesimpulan dari pengujian sistem tersebut.

5. KESIMPULAN

Kesimpulan dari hasil penggunaan metode *Equivalence Partitioning*. Didapatkan bahwa teknik sangat cocok pada metode dengan membagi ruang menjadi pilihan. Dalam hal ini, perangkat lunak dalam pengujian dapat berjalan optimal dengan metode *Equivalence Partitioning* sebagai pengujian *black box* sistem informasi.

REFERENCES

- Matteson, S. (2016). *Software. Report: Software failure caused \$1.7 trillion in financial losses in 2017*.
- Maulana, A., Kurniawan, A., Keumala, W., Sukma, V. R., & Saifudin, A. (2020). Pengujian Black Box pada Aplikasi Penjualan Berbasis Web Menggunakan Metode Equivalents Partitions (Studi Kasus: PT Arap Store). *Jurnal Teknologi Sistem Informasi dan Aplikasi*, 50-56.
- Ningrum, F. c., Suherman, D., Aryanti, S., Prasetya, H. A., & Saifudin, A. (2019). Pengujian Black Box pada Aplikasi Sistem Seleksi Sales Terbaik Menggunakan Teknik Equivalence Partitions. *Jurnal Informatika Universitas Pamulang*, 125-130.
- Nurudin, M., Jayanti, W., Saputro, R. D., Saputra, M. P., & Yulianti, Y. (2019). Pengujian Black Box pada Aplikasi Penjualan Berbasis Web Menggunakan Teknik Boundary Value Analysis. *Jurnal Informatika Universitas Pamulang*, 143.
- Matteson, S. (2018). *Report: Software failure caused \$1.7 trillion in financial losses in 2017*. <https://www.techrepublic.com/article/report-software-failure-caused-1-7-trillion-in-financial-losses-in-2017/>