

# Analisis Sistem Keuangan Desa (SISKEUDes) di Kecamatan Muara Sugihan Menggunakan Metode *Black Box Testing*

## VILLAGE FINANCIAL SYSTEM ANALYSIS (SISKEUDES) IN MUARA SUGIHAN DISTRICT USING BLACK BOX TESTING METHOD

Fadhila Tangguh Admojo<sup>1</sup>, M. Leo Adi Saputra<sup>2</sup>

*Program Studi Informatika: Institut Teknologi dan Bisnis PalComTech  
Jl. Basuki Rahmat No. 05, Palembang 30129, Indonesia  
E-mail: leopti02@gmail.com*

### Abstrak

Analisis Sistem Keuangan Desa (SISKEUDes) di Kecamatan Muara Sugihan Menggunakan Metode *Black Box Testing*. Aplikasi Sistem Keuangan Desa (SISKEUDes) dikembangkan bersama antara badan pengawasan keuangan dan pembangunan (BPK) dengan kementerian dalam negeri (Kemendagri) nomor 20 tahun 2018 tentang pengelolaan keuangan desa, 22 (dua puluh dua) desa di Kecamatan Muara Sugihan selalu menggunakan aplikasi Siskeudes dalam pengelolaan keuangan desa. Untuk memastikan aplikasi sesuai kebutuhan fungsional yang diharapkan maka diperlukan pengujian terhadap aplikasi Siskeudes, pengujian menggunakan *black box testing* yaitu melakukan validasi *output* dari data *input* yang diberikan. Dalam menentukan *test case* perlu adanya skenario *test case* untuk menentukan proses pengujian dari dokumen *requirement* dokumen persyaratan. Desain *test case* yang dapat membantu dalam menemukan kesalahan aplikasi merupakan pertimbangan penting dalam pengujian aplikasi. Terdapat beberapa teknik pengujian *back bok* yang digunakan dalam menentukan *tese case* diantaranya *equivalence partitioning*, *boundary value analysis*, *robustness testing*, dan *requirment testing*. Dari keempat teknik penggabungan pengujian metode *black box* dapat menemukan kesalahan aplikasi secara efektif, evaluasi hasil pengujian ini dibuktikan dengan perhitungan *defect removal efesiensi*, dari 160 *test case* ditemukan 24 *defect* perhitungan *defect removal efesiensi* =  $(24/160) \times 100\%$ , =  $0,15 \times 100\%$  = 15%, dari *test case* yang dibangun mendapatkan hasil pengujian 15% dari *test case* yang tidak lolos uji.

**Kata kunci** — analisis, black box, siskeudes

### Abstract

Analysis of the Village Financial System (SISKEUDes) in Muara Sugihan District Using the Black Box Testing Method. The Village Financial System Application (SISKEUDes) was developed jointly between the Financial and Development Supervisory Agency (BPK) and the Ministry of Home Affairs (Kemendagri) number 20 of 2018 regarding village financial management, 22 (twenty-two) villages in Muara Sugihan District always use the Siskeudes application in village financial management. To ensure the application meets the expected functional requirements, it is necessary to test the Siskeudes application, testing using black box testing is to validate the output of the input data provided. In determining the test case, it is necessary to have a test case scenario to determine the testing process from the requirements document. Test case design that can assist in finding application errors is an important consideration in application testing. There are several back-box testing techniques used in determining the test case, including equivalence partitioning, boundary value analysis, robustness testing, and requirement testing. Of the four combined black box methods testing techniques can find application errors effectively, the evaluation of the test results is evidenced by the calculation of defect removal efficiency, from 160 test cases found 24 defects calculation of defect removal efficiency =  $(24/160) \times 100\%$ , =  $0,15 \times 100\%$  = 15%, from the test cases that were built, 15% of the test cases did not pass the test.

**Kata kunci** — analisis, black box, siskeudes

### 1. PENDAHULUAN

Testing merupakan aktivitas yang digunakan untuk dapat melakukan evaluasi suatu parameter ataupun kemampuan dari program atau *sistem* serta memastikan apakah sudah memenuhi kebutuhan ataupun hasil yang diharapkan (Hetzel, 1983)[1]. Pengujian

perangkat lunak dilakukan agar bisa memberikan jaminan kualitas perangkat lunak dan tidak ada kesalahan pada *sistem* hal ini dilakukan agar semua bagian *sistem* dapat berfungsi tanpa adanya kesalahan dan sesuai dengan kebutuhan (fungsional) pengguna. Di daerah tempat saya tinggal terdapat 22 (dua puluh dua) Desa yang berada di otonomi daerah Kecamatan Muara Sugihan Kabupaten Banyuasin dan desa-desa tersebut dalam pengelolaan keuangan menggunakan aplikasi Sistem Keuangan Desa (Siskeudes).

Desa yang mengimplementasikan aplikasi Siskeudes terdiri dari desa, Daya Murni, Daya Bangun Harjo, Sumber Mulyo, Margo Mulyo 16, Sugih Waras, Indra Pura, Sido Makmur, Ganesha Mukti, Margo Rukun, Daya Kesuma, Tirtaharja, Rejosari, Argo Mulyo, Cendana, Mekar Jaya, Tirta Mulya, Beringin Agung, Jalur Mulya, Timbul Jaya, Gilirang, Juru Taro, Kuala Sugihan. Aplikasi Siskeudes memiliki 5 (lima) menu File, Parameter, Data Entri, Laporan, Tools, dan Help. Catatan release Siskeudes terbaru versi 2.0 R2.0.4 dirilis pada tanggal 20 September 2021.

Pengujian perangkat lunak dilakukan agar bisa memberikan jaminan kualitas dari perangkat lunak dan tidak ada kesalahan pada sistem. Dilakukan agar semua bagian sistem dapat berfungsi tanpa adanya kesalahan dan sesuai dengan kebutuhan pengguna serta memiliki kualitas produktivitas yang tinggi. Pengujian yang akan digunakan pada aplikasi Siskeudes adalah metode *black box testing*. Pemilihan metode *black box* focus kepada kebutuhan fungsional dari aplikasi, seseorang penguji dapat mendefinisikan *test case* dan melakukan evaluasi pada kebutuhan fungsional aplikasi untuk mencari kesalahan dalam beberapa kategori yaitu fungsi yang tidak benar, kesalahan antarmuka, kesalahan struktur data, kesalahan performansi, dan kesalahan inisialisasi maupun terminasi. *black box testing* memiliki banyak macam pengujian salah satunya adalah *equivalence partitioning* (partisi ekuivalensi), *boundary value analysis*, (analisa nilai batas), *robustness testing* (pengujian acak), dan *requirements testing* (pengujian kebutuhan) (Pressman, 2005)[2].

Metode pengujian *black box* telah menghasilkan analisa yang menyatakan bahwa perangkat lunak tidak secara keseluruhan menghasilkan produk yang sesuai dengan persyaratan fungsional. Berdasarkan hasil wawancara bersama bapak Sarju Sasono selaku staf Kecamatan Muara Sugihan bahwa aplikasi Siskeudes sangat penting bagi 22 (duapuluh dua) desa di Muara Sugihan dalam menata kelola keuangan desa, dari penginputan data manual menggunakan *Microsoft Excel* dan *Microsoft Word* sekarang sudah dimudahkan dalam penginputan data secara otomatis, sehingga memudahkan aparatur pemerintah desa dalam mengelola keuangan dan pelaporan, berjalan sejak tahun 2017 sampai sekarang.

Berdasarkan hasil analisis dan pentingnya aplikasi Siskeudes bagi desa diperlukannya pengujian tentang “**Analisis Sistem Keuangan Desa (SISKEUDES) di Kecamatan Muara Sugihan Menggunakan Metode Black Box Testing**”. Pertimbangan dipilihnya aplikasi Siskeudes di Kecamatan Muara Sugihan dikarenakan perangkat lunak tersebut selalu digunakan oleh operator desa dan instansi Kecamatan dalam pengumpulan data laporan setiap tahunnya. Hasil akhir penelitian ini adalah sejauh mana metode *black box* mampu diuji untuk mengetahui kesalahan-kesalahan dalam aplikasi Siskeudes. Hasil tersebut dapat digunakan oleh instansi terkait sebagai evaluasi dalam perilsan aplikasi selanjutnya.

## 2. METODE PENELITIAN

### Skenario Test Case

Tahap ini meliputi membuat skenario dari masing-masing *flowchart* modul, membuat *test case*, desain nilai *test case* menggunakan *Equivalence partitioning*, *boundary value analysis*, *robustness testing*, *requirement testing*. Selanjutnya *test case* akan berlanjut kepada tahap skenario *robustness testing* yang dibuat sesuai kebutuhan kasus pengujian yang diturunkan, pengujian berupa data acak berupa angka, huruf, dan symbol-simbol untuk menguji ketahanan dari perangkat lunak tersebut, setelahnya pengujian berlanjut pada *requirement testing* melalui hasil

sekenario pengamatan yang sudah berjalan dan mengelompokkan *test case* sesuai kebutuhan, untuk dilaksanakannya pengujian.

#### **Tahap Pengujian Equivalence Partitioning**

Tahap pengujian ini bertujuan untuk menentukan keadaan valid atau invalid terhadap kondisi input perangkat lunak. Pengujian ini berusaha menemukan sebuah *test case* yang mampu mengungkap kelas-kelas kesalahan, sehingga mengurangi jumlah total *test case* yang harus dikembangkan. Dalam tahap pengujian *equivalence partitioning* hal-hal yang dilakukan adalah Membuat daftar *test case* berdasarkan fungsi-fungsi yang ada dalam modul yang diuji. Daftar *test case* berisi tentang objek-objek pada modul yang hendak diuji. Membuat model komponen pengujian yang merupakan partisi dari nilai *input* dan *output* berdasarkan spesifikasi fungsi dan objek modul yang diuji. Melakukan pengujian berdasarkan model partisi yang dibuat.

#### **Tahap Pengujian Boundary Value Analysis**

Tahap pengujian ini bertujuan untuk menguji kemampuan program dalam menangani data pada batas yang dapat diterima perangkat lunak. Pengujian *boundary value analysis* merupakan komplemen dari teknik pengujian *equivalence partitioning* namun lebih pada memilih elemen-elemen di dalam kelas ekuivalen pada bagian nilai batas dari kelas tersebut. Membuat daftar *test case* berdasarkan fungsi-fungsi yang ada dalam modul yang diuji. Daftar *test case* berisi tentang objek-objek pada modul yang hendak diuji berdasarkan hasil pengujian *equivalence partitioning*. Daftar *test case* akan digunakan dalam pengujian selanjutnya. Membuat komponen pengujian yang merupakan hasil dari partisi dari pengujian *equivalence partitioning*, sehingga tahap pembuatan model harus menunggu pengujian *equivalence partitioning* selesai. Melakukan pengujian berdasarkan model *boundary value analysis* (BVA) yang dibuat.

#### **Tahap Pengujian Robusness Testing**

Pengujian kotak hitam *robustness testing* dengan memasukkan nilai-nilai yang diluar dari kebutuhan yang ditentukan (requisiten), tujuan untuk membuktikan bahwa tidak ada event yang kacau pada perangkat lunak dengan dimasukkannya nilai-nilai yang tidak normal, jika masukan sebuah password valid adalah 5 karakter [a..z:A...Z] kasus uji : masukkan sembarang karakter dan sembarang jumlah karakter, misal : z##12 atau iu831280j eksekusi *result* aplikasi Siskeudes tidak dapat menyimpan data.

#### **Tahap Pengujian Robusness Testing**

Pada tahap ini pengujian didefinisikan pada fase analisis terhadap requirement yang telah disusun pada *test case*, pengujian dilakukan pada kebutuhan fungsional requirement pada aplikasi Siskeudes sampai pada tahap cetak dokumen laporan, dari sekenario *test case* yang telah dibuat akan dikelompokkan sesuai dengan requirement pada modul.

#### **Evaluasi Hasil Pengujian**

Tahap ini melakukan proses evaluasi untuk memeriksa keefektifan metode dalam pengujian aplikasi. Evaluasi dapat dilakukan dengan menghitung *defect removal efficiency* yang dapat dilihat pada Persamaan  $DRE = (E/D) \cdot 100\%$  E= jumlah kesalahan yang ditemukan dan D= jumlah keseluruhan *test case*.

### **3. HASIL DAN PEMBAHASAN**

#### **Pengujian Black Box**

Tahap ini melakukan pengelompokan pada *test case* dengan membuat sekenario *test case* sesuai requirement serta membuat kasus uji berdasarkan class requirement, pengujian dilakukan secara berurutan menggunakan metode *black bok* dari calass *equivalence partitioning* diturunkan kepada

class data selanjutnya yaitu, *boundary value* pengujian nilai batas, *robustness testing* pengujian data acak dan *requirement testing* pengujian requirement yang disediakan.

### Hasil pengujian menggunakan teknik *equivalence partitioning* dan *requirement*

Ditemukanya *defect* atau kesalahan dari 127 *test case* ditemukanya 13 *defect* atau tidak lolos uji. Analisis kesalahan dijabarkan berdasarkan pada tabel 1.

**Tabel 1.** Pengujian Equivalence dan Requirement

No	Requirement Modul	Skenario Submenu	Skenario Judul Test Case	Langkah-langkah	Data Tes	Hasil Yang Harapkan	Hasil Aktual	Status Berhasil/Gagal
1.	2.1.1.	File-Login	User login valid	Input data user ID, password dan tahun anggaran, klik ok	ID : 132014 Pass :132014 tahun :2022	Berhasil login, dapat mengoperasikan menu dan submenu	Berhasil	Berhasil
2.	2.1.1.	File-Login	User login tidak valid mengubah kode desa user ID dan password kode desa	Input data user ID, password dan tahun anggaran, klik ok	ID:132013 Pass :132013 dan ID:112014 Pass:112014 tahun :2022	Tidak berhasil login muncul notifikasi “ Password anda salah coba diulangi lagi” “ mohon maaf userID yang anda masukkan tidak terdaftar”	Berhasil	Gagal
3.	2.1.2.	Parameter-Rekening Bang Desa	User pilih desa kecamatan, kemudian klik rekening kas desa, user tidak dapat mengganti kode rincian rekening kas desa, rekening hanya disediakan satu kali input, user dapat klik tambah, ubah, hapus, batal, simpan dan tutup.	Klik menu parameter, double klik rekening bank desa, double klik kecamatan desa, double klik rekening kas desa, isi data sesuai ketentuan, klik simpan.	No Rek:16730105374Bank:Sumsel Babel Kantor Cbang:Pangkalan Balai Pemilik Rek:Ganesha Mukti Alamat:Desa Ganesha Mukti No Hp:082371438968 NIK:1607142106980003	Berhasil input data rekening bank desa dan dapat disimpan pada database.	Berhasil	Berhasil

No	Requirment Modul	Sekenario Submenu	Sekenario Judul Test Case	Langkah-langkah	Data Tes	Hasil Yang Harapkan	Hasil Aktual	Setatus Berhasil/Gagal
4.	2.1.2.	Parameter-Rekening Bang Desa	User pilih desa kecamatan, kemudian klik rekening kas desa, user tidak dapat mengganti kode rincian rekening kas desa, rekening hanya disediakan satu kali input, user dapat klik tambah, ubah, hapus, batal, simpan dan tutup . Input data tidak valid	Klik menu parameter, double klik rekening bank desa, double klik kecamatan desa, double klik rekening kas desa, isi data sesuai ketentuan, klik simpan.	No Rek:16730105379drjfk#RS No Hp:082371438968drjfk#RS NIK:160714210698 RSdrjfk#RS	Tidak berhasil input data rekening bank desa sistem menolak input dan tidak dapat disimpan pada database.	Berhasil	Gagal
5.	2.1.3.	Data Entri-Perencanaan-Data Umum RPJMDesaData Umum DesaPelaksana Kegiatan	User dapat melakukan input data dengan klik tambah data otomatis, maksimal no ID 99 (Sembilan puluh Sembilan), user dapat klik ubah, tambah, hapus, batal, simpan dan tutup. Data valid.	Klik menu data entri, klik data umum RPJM Desa, klik data umum desa, input data sesuai ketentuan, jika sudah klik simpan	No. ID:01 Nama:M Yunus Jabatan:Kepala Desa	Data berhasil ditambahkan dapat menjalankan semua fungsi tombol,	Berhasil	Berhasil
6.	2.1.3.	Data Entri-Perencanaan-Data Umum RPJMDesaData Umum DesaPelaksana Kegiatan	User dapat melakukan input data dengan klik, maksimal no ID 99 (Sembilan puluh Sembilan), user dapat klik ubah, tambah, hapus, batal, simpan dan tutup.	Klik menu data entri, klik data umum RPJM Desa, klik data umum desa, input data sesuai ketentuan, jika sudah klik simpan	No. ID:01.99,89,56. Nama:M Yunus Jabatan:Kepala Desa	Data berhasil ditambahkan dapat menjalankan semua fungsi tombol,	Berhasil	Gagal

No	Requirment Modul	Sekenario Submenu	Sekenario Judul Test Case	Langkah-langkah	Data Tes	Hasil Yang Harapkan	Hasil Aktual	Setatus Berhasil/Gagal
			Data tidak valid.					
7.	2.1.3.	Data Entri-Perencanaan-Data Umum RPJMDesaRPJMDesa-BidangKegiatan-Dana Indikatif	User dapat melakukan input data dengan klik tambah isi data dengan sesuai keterangan yang diminta RAB, bagian Sas hanya dapat input angka, user dapat melakukan tambah data, cetak, ubah, hapus, batal, simpan dan tutup. Data valid	Klik menu data entri, klli data umum RPJM Desa, klik RPJM Desa, klik bidang dan pilih, klik sub bidang dan pilih, klik kegiatan input data sesuai ketentuan, klik dana indikatif, klik dana infikatif, klik tambah tambah.	Tahun:1 Lokasi:Desa Ganesha Mukti Volume:1 tahun Sas:1 orang Pelaksana:guru ngaji Sumber dana:DDS Biaya:1000.0000 Waktu 12 bulan Tgl:10/01/2022 Keg:Swakelola	Data berhasil disimpan, user dapat menjalankan semua fungsi tombol.	Berhasil	Berhasil

**Tabel 2.** Pengujian Boundary Value Analysis

No	Requirment Modul	Sekenario Submenu	Sekenario Judul Test Case	Langkah-langkah	Data Tes	Hasil Yang Harapkan	Hasil Aktual	Setatus Berhasil/Gagal
11.	2.1.3.	Data Entri-Penganggaran -Isian Data AnggaranKegiatan-BidangSub Bidang-kegiatan	Pengujian pagu anggaran, nilai batas maksimum 1 M dan minimum 0	Klik data entri, penganggaran, klik isian data anggaran, klik pendapatan, kegiatan	A:-82,394,928 B:-999,999,999,999,999 C:999,999,999,999,999	Tidak dapat menyimpan data - 1, tidak dapat menyimpan data melebihi 1 Miliar	Berhasil	Gagal
12.	2.1.3.	Data Entri-Penganggaran -Isian Data AnggaranPendapatan-RAP-RAP Rinci	Pengujian pada bagian Harga Satuan (Rp), nilai batas maksimum 1 M dan minimum 0	Klik data entri, penganggaran, klik isian data anggaran, klik Pendapatan, RAP rinci	A:-999999999999999	Tidak dapat menyimpan data > 1 M dan < 0.	Berhasil	Gagal
13.	2.1.3.	Data Entri-Penganggaran -Isian Data AnggaranBelanja-BidangKegiatan-Keg SubRAB-RAB Rinci	Pengujian pada bagian Harga Satuan (Rp), nilai batas maksimum 1 M dan minimum 0	Klik data entri, penganggaran, klik isian data anggaran, klik belanja, pilih bidang, pilih kegiatan, klik	A:-2,500,000	Tidak dapat menyimpan data > 1 M dan < 0. muncul notifikasi “ada kelebihan belanja yang melebihi realisasi”	Berhasil	Gagal

No	Requirement Modul	Skenario Submenu	Skenario Judul Test Case	Langkah-langkah	Data Tes	Hasil Yang Harapkan	Hasil Aktual	Status Berhasil/Gagal
				keg sub, klik RAB, klik RAB rinci				
14.	2.1.3.	Data Entri-Penganggaran -Isian Data AnggaranPembayaan 1-RAPRAP Rinci	Pengujian pada bagian Harga Satuan (Rp), nilai batas maksimum 1 M dan minimum 0	Klik data entri, penganggaran, klik isian data anggaran, klik pembiayaan 1, klik RAB, klik RAB rinci	A:-32,912,863	Tidak dapat menyimpan data > 1 M dan < 0. muncul notifikasi “ada kelebihan belanja yang melebihi realisasi”	Berhasil	Gagal

**Tabel 3.** Pengujian Robustness Testing

No	Requirement Modul	Skenario Submenu	Skenario Judul Test Case	Langkah-langkah	Data Tes	Hasil Yang Harapkan	Hasil Aktual	Status Berhasil/Gagal
20.	2.1.3.	Data Entri-Penganggaran -Isian Data AnggaranKegiatan-BidangSub Bidang-kegiatan	Pengujian dilakukan pada bagian Pagu. Sistem akan menolak semua input simbol dan huruf.	Klik data entri, penganggaran, klik isian data anggaran, klik pendapatan, kegiatan	A:~!@#\$%^&*()_+={} []"/- B:abcdefghijklmnopqrstu vwxyz	Tidak dapat menyimpan data, sistem menolak input.	Berhasil	Berhasil
21.	2.1.3.	Data Entri-Penganggaran -Isian Data AnggaranKegiatan-BidangSub Bidangkegiatan, paket kegiatan	Pengujian pada bagan Nilai (Rp). Pengujian dilakukan pada bagian Pagu. Sistem akan menolak semua input simbol dan huruf.	Klik data entri, penganggaran, klik isian data anggaran, klik pendapatan, kegiatan, klik Paket Kegiatan	A:~!@#\$%^&*()_+={} []"/- B:abcdefghijklmnopqrstu vwxyz	Tidak dapat menyimpan data, sistem menolak input.liar	Berhasil	Berhasil
22.	2.1.3.	Data Entri-Penganggaran -Isian Data AnggaranPendapatan-RAP-RAP Rinci	Pengujian dilakukan pada bagian harga satuan dan jumlah satuan. Sistem akan menolak semua input simbol dan huruf.	Klik data entri, penganggaran, klik isian data anggaran, klik Pendapatan, RAP rinci	A:~!@#\$%^&*()_+={} []"/- B:abcdefghijklmnopqrstu vwxyz	Tidak dapat menyimpan data, sistem menolak input.liar	Berhasil	Berhasil

## Hasil Pengujian

Berdasarkan hasil pengujian *black bok* akan dilakukanya perbandingan dari *test case* yang telah dibangun dengang *defect* atau kesalahan yang ditemukan pada aplikasi Siskeudes pada bagian kesalahan yang ditemukan, berikut perbandingan pengujian dari ke 4 (empat) teknik *black box*.

## Hasil Pengujian *Equivalence Partitioning* dan *Requirement*

Hasil pengujian menggunakan teknik *equivalence partitioning* dan *requirement*, ditemukanya *defect* atau kesalahan dari 127 *test case* ditemukanya 13 *defect* atau tidak lolos uji. Analisisi kesalahan dijabarkan berdasarkan pada tabel 4.

**Tabel 4** Temuan Equivalence dan Requirement

No	Dokumen Requirement	Hasil Pengujian
4.4.1.	Login	Gagal
4.4.2.	Parameter-Rekening Bank Desa-No Rek	Gagal
4.4.2.	Parameter-Rekening Bank Desa-No HP	Gagal
4.4.3.1	Data Entri-Perencanaan-Data Umum RPJMDes-Data Umum DesaPelaksana Kegiatan.	Gagal
4.4.3.1	Data Entri-Perencanaan-Data Umum RPJMDes-RPJMDes-Bidang- Kegiatan-Dana Indikatif.	Gagal
4.4.3.2	Data Entri-Penganggaran- Isian Data Anggaran-Kegiatan-Bidang-Sub Bidang-Kegiatan	Gagal
4.4.3.2	Data Entri-Penganggaran-Isian Data Anggaran-Kegiatan-Bidang-Sub Bidang-Kegiatan-Paket Kegiatan	Gagal
4.4.3.2	Data Entri-Penganggaran-Isian Data Anggaran-Belanja-Bidang-Keg SubRAB-RAB Rinci	Gagal
4.4.3.2	Data Entri-Penganggaran-Isian Data Anggaran-Pembiayaan 1-RAP-RAP Rinci	Gagal
4.4.3.2	Data Entri-Isian Data Anggaran-Pembiayaan 2-RAP-RAP Rinci	Gagal
4.4.3.3	Data Entri-Penatausahaan-Penerimaan Desa-Penerimaan Tunai-TBPRincian TBP.	Gagal
4.4.3.3	Data Entri-Penatausahaan-Penerimaan Desa-Penerimaan Bank-TBPRincian TBP	Gagal
4.4.3.3	Data Entri-Penatausahaan-SPP Kegiatan-SPP Definitif-SPP-Rincian SPPBukti Pengeluaran	Gagal

- 1) Pada tabel no 7 (tujuh) requirement 4.4.1. dari tahap pengujian ditemukanya kegagalan atau *defect* pada form *login*, analisis kesalahan berupa data input kode desa yang seharusnya *sistem* tidak dapat *login* namun *sistem* menerima *login*, berikut perbandingan data ID yang digunakan yaitu ID desa Ganesha Mukti "ID:132014 password:132014" *sistem* dapat *login* dengan sempurna, jika ID dan password berbeda maka *sistem* akan menolak dan muncul notifikasi "password anda salah coba diulangi lagi" "mohon maaf user ID yang anda masukkan tidak terdaftar". berikut data ID dan password desa yang berbeda "ID:132013 password:132013" "ID:112014 password:112014" dari data *input* tersebut *sistem* tidak menolak dan berhasil *login* namun data didalamnya kosong atau mengulang kembali penginputan data dari awal, hal ini dapat memicu kehilangan data bagi *user*.
- 2) Pada tabel no 13 (tiga belas) requirement 4.4.2. dari tahap pengujian ditemukanya kegagalan atau *defect* pada form rekening bank desa pada bagian no rekening desa yang seharusnya *sistem* tidak menerima nilai *input* huruf dan simbol namun *sistem* menerima data *input* tersebut, berikut perbandingan data, data valid "no rekening : 16730105379" data invalid "no rekenig : 16730105379drjfk#RS".
- 3) Pada tabel no 14 (empat belas) requirement 4.4.2. dari tahap pengujian ditemukanya kegagalan atau *defect* pada form rekening bank desa pada bagian no hp yang seharusnya *sistem* tidak menerima nilai *input* huruf dan simbol namun *sistem* menerima data *input* tersebut, berikut perbandingan data, data valid "no hp: 082371143896" data invalid "no hp: 082371143896"
- 4) Pada tabel no 17 (tujuh belas) requirement 4.4.3.1. dari tahap pengujian ditemukanya kegagalan atau *defect* pada form pelaksana kegiatan dibagian no ID yang seharusnya tidak dapat diubah dan nomer berurutan, berikut data no ID tidak valid "01,99,89,56".



- 5) Pada tabel no 21 (dua puluh satu) requirement 4.4.3.1. dari tahap pengujian ditemukanya kegagalan atau *defect* pada form dana indikatif dibagian biaya, *sistem* tidak dapat menerima nilai *input* berupa huruf dan simbol, berikut data invalid dari *input* data “biaya: 1E201” hanya huruf (E) dari pengujian huruf dan simbol yang dapat disimpan.
- 6) Pada tabel no 25 (dua puluh lima) requirement 4.4.3.2. dari tahap pengujian ditemukanya kegagalan atau *defect* pada form kegiatan kegiatan dibagian pagu, *sistem* tidak dapat menerima nilai *input* berupa huruf dan simbol, berikut data invalid dari *input* data “pagu : 4,55E272” hanya huruf (E) dari pengujian huruf dan simbol.
- 7) Pada tabel no 27 (dua puluh tujuh) requirement 4.4.3.2. dari tahap pengujian ditemukanya kegagalan atau *defect* pada form paket kegiatan kegiatan dibagian nilai (RP), *sistem* tidak dapat menerima nilai *input* berupa huruf dan simbol, berikut data invalid dari *input* data “pagu : 4,2332E27” hanya huruf (E) dari pengujian huruf dan simbol.
- 8) Pada tabel no 30 (tiga puluh) requirement 4.4.3.2. dari tahap pengujian ditemukanya kegagalan atau *defect* pada form bidang RAB rinci dibagian no urut, no urut pada *sistem* tidak dapat dirubah dan nomor dari data yang disimpan berurutan, seperti “no urut:45,01,88”.
- 9) Pada tabel no 32 (tiga puluh dua) requirement 4.4.3.2. dari tahap pengujian ditemukanya kegagalan atau *defect* pada form pembiayaan 1 RAB rinci dibagian no urut, no urut pada *sistem* tidak dapat dirubah dan nomor dari data yang disimpan berurutan, seperti “no urut:03,45,88”.
- 10) Pada tabel no 34 (tiga puluh empat) requirement 4.4.3.2. dari tahap pengujian ditemukanya kegagalan atau *defect* pada form pembiayaan 2 RAB rinci dibagian no urut, no urut pada *sistem* tidak dapat dirubah dan nomor dari data yang disimpan berurutan, seperti “no urut:01,87,45”.
- 11) Pada tabel no 46 (empat puluh enam) requirement 4.4.3.3. dari tahap pengujian ditemukanya kegagalan atau *defect* pada form rincian TBP dibagian nilai, *sistem* tidak dapat menerima nilai *input* berupa huruf dan simbol, berikut data invalid dari *input* data “pagu: 2,5E48”, dan tidak dapat cetak print previwe halaman.
- 12) Pada tabel no 49 (empat puluh sembilan) requirement 4.4.3.3. dari tahap pengujian ditemukanya kegagalan atau *defect* pada form penerimaan bank rincian TBP dibagian nilai, *sistem* tidak dapat menerima nilai *input* berupa huruf dan simbol, berikut data invalid dari *input* data “pagu : 891,87E,103”, dan tidak dapat cetak print previwe halaman.
- 13) Pada tabel no 56 (lima puluh enam) requirement 4.4.3.3. dari tahap pengujian ditemukanya kegagalan atau *defect* pada form SPP definitif bukti pengeluaran dibagian no rek dan NPWP, *sistem* tidak dapat menerima nilai *input* berupa huruf dan simbol, berikut data invalid dari *input* data “rek bank :1673xyz%#\$10537” dan “NPWP: 9\$24xyj54o31 4ooo”.

### Hasil Pengujian Boundary Value Abalysis

Hasil pengujian menggunakan teknik *boundary value* ditemukanya 11 *defect* atau kesalahan. Analisisi kesalahan dijabarkan berdasarkan pada tabel 5

**Tabel 5.** Hasil Pengujian Boundary Value Analysis

No	Dokumen Requirement	Hasil Pengujian
4.4.3.2.	Data Entri-Penganggaran-Isian Data Anggaran-Kegiatan-Bidang-Sub Bidang-Kegiatan	Gagal
4.4.3.2.	Data Entri-Penganggaran-Isian Data Anggaran-Kegiatan-Bidang-Sub Bidang-Kegiatan-Paket Kegiatan	Gagal
4.4.3.2.	Data Entri-Penganggaran-Isian Data Anggaran-Pendapatan-RAP-PAP Rinci	Gagal
4.4.3.2.	Data Entri-Penganggaran-Isian Data Anggaran-Belanja-BidangKegiatan-Keg Sub-RAB-RAB Rinci	Gagal
4.4.3.2.	Data Entri-Penganggaran-Isian Data Anggaran-Pembiayaan 1-RAPRAP Rinci	Gagal
4.4.3.2.	Data Entri-Penganggaran-Isian Data Anggaran-Pembiayaan 2-RAPRAP Rinci	Gagal
4.4.3.3.	Data Entri-Penatausahaan-SPP Kegiatan-SPP Definitif-SPP-Rincian SPP-Bukti Pengeluaran-Potongan	Gagal
4.4.3.3.	Data Entri-Penatausahaan-Pengembalian-Belanja-Bukti PengembalianRincian Bukti.	Gagal
4.4.3.3.	Data Entri-Penatausahaan-Mutasi Kas-Penyetoran-Mutasi Kas	Gagal
4.4.3.3.	Data Entri-Penatausahaan-Mutasi Kas-pendapatan Bunga-Mutasi Kas.	Gagal
4.4.3.3.	Data Entri-Penata Usahaan-Mutasi Kas-Biaya Admin Bank-Mutasi Kas	Gagal

- 1) Pada tabel no 128 (seratus dua puluh delapan) requirement 4.4.3.2. dari tahap pengujian ditemukan kegagalan atau *defect* pada bagian pagu, *sistem* seharusnya tidak dapat menyimpan nilai kurang dari (0) namun *sistem* menerima input data (-1) dan seterusnya, berikut data uji tidak valid (a:-82,394,928 / b:-999,999,999,999,999) data *input* melebihi nilai batas minimum.
- 2) Pada tabel no 129 (seratus dua puluh sembilan) requirement 4.4.3.2. dari tahap pengujian ditemukan kegagalan atau *defect* pada paket kegiatan bagian nilai, *sistem* seharusnya tidak dapat menyimpan nilai kurang dari (0) namun *sistem* menerima input data (-1) dan seterusnya, berikut data uji tidak valid (a:-8,888,888,888,888) data *input* melebihi nilai batas minimum.
- 3) Pada tabel no 130 (seratus tiga puluh) requirement 4.4.3.2. dari tahap pengujian ditemukan kegagalan atau *defect* pada pendapatan RAB rinci bagian harga satuan, *sistem* seharusnya tidak dapat menyimpan nilai kurang dari (0) namun *sistem* menerima input data (-1) dan data dapat disimpan, berikut data uji tidak valid (a:-999,999,999,999,999) data *input* melebihi nilai batas minimum.
- 4) Pada tabel no 131 (seratus tiga puluh satu) requirement 4.4.3.2. dari tahap pengujian ditemukan kegagalan atau *defect* pada kegiatan RAB rinci bagian harga satuan, *sistem* seharusnya tidak dapat menyimpan nilai kurang dari (0) namun *sistem* menerima input data (1) dan data dapat disimpan, berikut data uji tidak valid (a:-2,500,000) data *input* melebihi nilai batas minimum.
- 5) Pada tabel no 132 (seratus tiga puluh dua) requirement 4.4.3.2. dari tahap pengujian ditemukan kegagalan atau *defect* pada pembiayaan 1 RAP rinci bagian harga satuan, *sistem* seharusnya tidak dapat menyimpan nilai kurang dari (0) namun *sistem* menerima input data (1) dan data dapat disimpan, berikut data uji tidak valid (a:-32,912,863) data *input* melebihi nilai batas minimum.
- 6) Pada tabel no 133 (seratus tiga puluh tiga) requirement 4.4.3.2. dari tahap pengujian ditemukan kegagalan atau *defect* pada pembiayaan 2 RAP rinci bagian harga satuan, *sistem* seharusnya tidak dapat menyimpan nilai kurang dari (0) namun *sistem* menerima input data (1) dan data dapat disimpan, berikut data uji tidak valid (a:-35,900,000) data *input* melebihi nilai batas minimum.

- 7) Pada tabel no 139 (seratus tiga puluh sembilan) requirement 4.4.3.3. dari tahap pengujian ditemukan kegagalan atau *defect* pada bukti pengeluaran potongan bagian nilai, *sistem* seharusnya tidak dapat menyimpan nilai kurang dari (0) namun *sistem* menerima input data (1) dan data dapat disimpan, berikut data uji tidak valid (nilai:-4,800,000) data *input* melebihi nilai batas minimum.
- 8) Pada tabel no 141 (seratus empat puluh satu) requirement 4.4.3.3. dari tahap pengujian ditemukan kegagalan atau *defect* pada rincian bukti bagian nilai, *sistem* seharusnya tidak dapat menyimpan nilai kurang dari (0) namun *sistem* menerima input data (-1) dan data dapat disimpan, berikut data uji tidak valid (nilai:-1,000,000) data *input* melebihi nilai batas minimum.
- 9) Pada tabel no 142 (seratus empat puluh dua) requirement 4.4.3.3. dari tahap pengujian ditemukan kegagalan atau *defect* pada penyetoran mutasi kas bagian nilai, *sistem* seharusnya tidak dapat menyimpan nilai kurang dari (0) namun *sistem* menerima input data (-1) dan data dapat disimpan, berikut data uji tidak valid (nilai:-30,463,561) data *input* melebihi nilai batas minimum.
- 10) Pada tabel no 143 (seratus empat puluh tiga) requirement 4.4.3.3. dari tahap pengujian ditemukan kegagalan atau *defect* pada pendapatan bunga mutasi kas bagian nilai, *sistem* seharusnya tidak dapat menyimpan nilai kurang dari (0) namun *sistem* menerima input data (1) dan data dapat disimpan, berikut data uji tidak valid (nilai:-34,566) data *input* melebihi nilai batas minimum.
- 11) Pada tabel no 144 (seratus empat puluh empat) requirement 4.4.3.3. dari tahap pengujian ditemukan kegagalan atau *defect* pada biaya admin bank mutasi kas bagian nilai, *sistem* seharusnya tidak dapat menyimpan nilai kurang dari (0) namun *sistem* menerima input data (1) dan data dapat disimpan, berikut data uji tidak valid (nilai:-3,000 data *input* melebihi nilai batas minimum.

### Evaluasi Hasil Pengujian

Berdasarkan hasil pengujian dari ke 4 (empat) teknik *black bok* dengan pengujian 160 *test case* ditemukan 24 *defect*, dari hasil pegujian *black box* akan dievaluasi dengan membagi jumlah *defect* ditemukan dan jumlah *test case* yang dibangun menggunakan rumus DRE, rincian tabel dari ke 4 (empat) teknik *black box* dapat dilihat pada tabel 5.

Tabel 5. Evaluasi Hasil Pengujian

Evalusi Pengujian Black Box	Test Case	Defect
Equivalence Partitioning dan Requirements Testing	127	13
Boundary Value Anlysis	17	11
Robustness Testing	16	0
Jumlah	160	24

Berikut perhitungan menggunakan rumus  $DRE = \frac{\text{jumlah kesalahan yang ditemukan}}{\text{jumlah keseluruhan test case}} \times 100\%$  untu memeriksa keefektifan metode dalam pengujian aplikasi.

$$DRE = \frac{E}{E+D} \times 100\%$$

$$DRE = \frac{160}{160+24} \times 100\%$$

$$DRE = \frac{160}{184} \times 100\%$$

$$DRE = 0,15 \times 100\%$$

$$DRE = 15\%$$

Dari perhitungan DRE diperoleh hasil keefektifan metode sebesar 15% dari total 160 *test case* yang diujikan dan 24 *defect* yang ditemukan.

#### 4. KESIMPULAN

Berdasarkan hasil dan analisa pengujian aplikasi Sistem Keuangan Desa (SISKEUDes) di Kecamatan Muara Sugihan menggunakan metode *Black Bo Testing* dapat disimpulkan :

- 1) Berdasarkan hasil pengujian *black box* dari 160 *test case* ditemukanya 24 *defect*, sehingga didapatkan nilai  $DRE = (24/160) * 100\% = 0,15$   $DRE = 0,15 * 100\% = 15\%$ . Hal ini menunjukkan metode *black bok* dapat menemukan kesalahan atau *defect* pada aplikasi Siskeudes sebesar 15% .
- 2) Setelah melakukan pengujian diketahui bahwa fungsionalitas masih bisa berjalan namun masih dapat menerima masukan data yang tidak diharapkan sehingga menyebabkan data yang disimpan tidak valid.
- 3) Dari ke 4 (empat) teknik *black box* yaitu *equivalence parititioning*, *boundary balue*, dan *requirement*, mampu mengungkap kesalahan pada aplikasi Siskeudes, namun pada pengujian *robustness* tidak ditemukanya kesalahan.

#### DAFTAR PUSTAKA

- [1] G.D. Everett, "*Software Testing Software Development Life Cycle*," Canada: IEEE Press, 2007.
  - [2] J. Tian, "*Software Quality Engineering*," Canada: Wiley, 2005.
  - [3] P. Ron, "*Software Testing*," United States of America: SAMS Publishing, 2004.
  - [4] G. Fournier, "*Essential Software Testing A Use-Case Approach*," New York: CRC Press, 2009.
  - [5] Tri Snadhika Jaya, "*Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung)*," (2018). *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, Vol.03, No.02, Januari 2018, ISSN: 2477-5126, e-ISSN: 2548-9356.
  - [6] Taufik Hidayat, Hendar Dini Putri, "*Pengujian Portal Mahasiswa pada Sistem Informasi Akademik (SINA) menggunakan Black Box Testing dengan Metode Equivalence Partitioning dan Boundary Value Analysis*," (2019). *JUTIS* Vol. 7 No. 1 Bulan April 2019, p-ISSN: 2252-5351, e-ISSN: 2656-0860.
  - [7] S. Ikhlashi and H. P. Purto, "*Komparasi Dua Teknik Black Box Testing: Equivalence Partitioning dan Boundary Value Analysis*," *Annu. Res. Semin.* 2019, vol. 5, no. 1. pp. 213-220, 2019.
-