

# PENGUJIAN SISTEM PENDUKUNG KEPUTUSAN PENENTUAN JURUSAN PADA SISWA SMA DENGAN MENGGUNAKAN METODE *BLACK BOX* BERBASIS *EQUIVALENCE PARTITIONS*

Khonsa Salsabila<sup>1</sup>, Fetty Tri Anggraeny<sup>2</sup>, Agung Mustika Rizki<sup>3</sup>

<sup>1,2</sup> Informatika, Fakultas Ilmu Komputer, <sup>3</sup> Universitas Pembangunan Nasional “Veteran” Jawa Timur

<sup>1</sup> salsabilakhonsa043@gmail.com, <sup>2</sup> fettyanggraeny.if@upnjatim.ac.id, <sup>3</sup> agung.mustika.if@upnjatim.ac.id

---

## Abstrak

Pengujian kualitas pada perangkat lunak sangat penting sebelum pengguna menggunakan perangkat lunak tersebut. Pengujian terhadap perangkat lunak diperlukan untuk meminimalisasi kerugian yang terjadi akibat kesalahan sistem. Pada penelitian ini yaitu pengujian pada perangkat lunak sistem pendukung keputusan penentuan jurusan pada siswa SMA dengan menggunakan *Black Box Testing* berbasis *Equivalence Partitions*. *Black Box Testing* sendiri adalah pengujian pada perangkat lunak tanpa harus memperhatikan detail perangkat lunak tersebut. Ada beberapa cara dalam *Black Box Testing* salah satunya *Equivalence Partitions*, *Equivalence Partitions* ini berfokus pada pengujian dengan menggunakan masukan data pada masing-masing *form* yang terdapat pada sistem pendukung keputusan penentuan jurusan, setiap *form* masukan akan dilakukan pengujian dan dikelompokkan berdasarkan fungsinya baik itu sesuai maupun tidak. Tahapan penelitian yang dilakukan dimulai dengan menentukan fungsionalitas yang akan diuji, merancang skenario pengujian, menentukan data yang akan diuji, menentukan *input* dapat berupa nilai numerik, rentang nilai dll sesuai dengan struktur basis data yang telah dibuat, melakukan pengujian eksperimen, mendokumentasikan hasil penelitian, dan menggambar kesimpulan. Hasil dari pengujian dengan menggunakan *Black Box Testing* berbasis *Equivalence Partitions* memberikan hasil dimana kita akan mengetahui apakah perangkat lunak ini memiliki kesalahan atau tidak, serta pengujian ini dapat menjamin kualitas dari perangkat lunak sistem pendukung keputusan penentuan jurusan pada siswa SMA ini.

**Kata kunci** : Penjurusan SMA, *Black Box Testing*, *Equivalence Partitions*

---

## 1. Pendahuluan

Pengujian terhadap perangkat lunak sangat penting dilakukan dengan tujuan untuk memperoleh jaminan kualitas perangkat lunak yang dihasilkan bebas dari terjadinya kesalahan (Debiyanti et al., 2020). *Software testing* merupakan cara eksekusi program untuk menemukan kesalahan. Pengujiannya bertujuan untuk mencari sebanyak mungkin kesalahan (*bug error*), serta untuk mengurangi resiko yang terkandung dalam suatu sistem komputer. Suatu pengujian dikatakan berhasil apabila menemukan kesalahan-kesalahan yang belum terdeteksi. Secara umum pengujian dilakukan untuk melihat selisih terhadap hasil yang diinginkan dengan hasil sesungguhnya. *Software testing* penting untuk dilakukan demi mengecek seluruh kesalahan yang terdapat pada program tersebut sehingga tidak memunculkan hal-hal yang merugikan saat digunakan (Ningrum et al., 2019).

Pada sistem pendukung keputusan penentuan jurusan pada siswa SMA ini merupakan proses pemilihan jurusan pada siswa SMA sesuai dengan minat, bakat serta kemampuan siswa tersebut. Sistem pendukung keputusan penentuan jurusan ini dikembangkan dengan tujuan untuk mengeluarkan

hasil yang dapat memberikan pertimbangan kepada pihak sekolah dalam menentukan jurusan pada siswa dengan menggunakan metode *Fuzzy-Preference Ranking Organization Method for Enrichment Evaluation* kepanjangan dari *Fuzzy-PROMETHEE*.

Metode *Black Box* merupakan metode yang digunakan untuk menguji sebuah perangkat lunak tanpa harus memperhatikan detail perangkat lunak tersebut. Pada pengujian *Black Box*, lamanya banyaknya tes data dapat diukur dari banyaknya masukan data pokok yang akan diuji, sistematis masukan yang harus terpenuhi selain itu batas masukan atas dan batas masukan bawah baik yang memenuhi spesifikasi. Tidak ada suatu maksud untuk mengetahui *source code* apa yang *output* pakai (Snadhika Jaya, 2018).

Ada beberapa cara dalam menguji *Black Box* salah satunya adalah dengan menggunakan teknik *Equivalence Partitions*. *Equivalence Partitions* merupakan *testing* dengan menggunakan masukan data pada masing-masing *form* yang terdapat pada sistem pendukung keputusan penentuan jurusan, setiap *form* masukan akan dilakukan pengujian dan dikelompokkan berdasarkan fungsinya baik itu sesuai maupun tidak sesuai (Hidayat & Muttaqin, 2018).

Berdasarkan *testing* yang dilakukan, kemudian akan diketahui kekurangan suatu sistem setelah dilakukan pengujian dengan menggunakan *Black Box* berbasis *Equivalence Partitions* dan dengan cara apa untuk mengetahui hasil yang dianggap sesuai. Oleh karena itu, tujuan dari pengujian ini yaitu untuk mengetahui kekurangan dari sistem supaya data yang dihasilkan selaras dengan data yang dimasukkan setelah data di proses, serta menghindari kesalahan pada sistem sebelum digunakan oleh pengguna (Hidayat & Muttaqin, 2018).

## 2. Landasan Teori

### 2.1 Pengujian Perangkat Lunak

Perangkat lunak memerlukan pengujian untuk memastikan perangkat lunak yang sudah atau sedang di buat dapat berjalan sesuai dengan *fungsiionalitas* yang diharapkan. *Developer* atau pengembang harus menyiapkan seperangkat uji coba untuk menguji program yang sudah selesai di buat agar kekurangan atau kesalahan dapat di deteksi sejak awal dan dilakukan perbaikan pada siklus berikutnya. Dalam melakukan *testing* yang harus diperhatikan adalah melakukan perancangan yang baik sehingga dapat dengan mudah menemukan kesalahan pada saat *testing*, sehingga dapat melakukan perbaikan dengan cepat dan menghemat waktu pengujian (Adi et al., 2020). *Testing* yaitu tahapan penting dari jaminan kualitas perangkat lunak dan juga tahapan yang tidak boleh dipisahkan dari siklus hidup pengembangan perangkat lunak sebagaimana analisis, desain dan pengkodean (Kesuma Jaya et al., 2019). Tujuan dari pengujian itu sendiri yaitu untuk memastikan perangkat lunak yang dibuat sesuai dengan kebutuhan yang telah ditentukan sebelumnya (Maulana et al., 2020). *Testing* yang dilakukan tidak lengkap dan efektif akan menimbulkan masalah dan menimbulkan kerugian ketika perangkat lunak tersebut digunakan (Trengginaz et al., 2020). Terdapat beberapa jenis *software testing*, antara lain (Utomo et al., 2020) :

- 1) Pengujian *White Box* yaitu poses pengujian yang berdasarkan pada pemeriksaan bagian detail perancangan, memakai struktur kontrol dari desain program secara prosedural untuk memecah pengujian ke dalam beberapa *case* pengujian. Sehingga kesimpulan dari pengertian *White Box Testing* yaitu petunjuk untuk memperoleh program yang benar secara 100%.
- 2) Pengujian *Black Box* yaitu pengujian yang menitikberatkan pada spesifikasi fungsional dari perangkat lunak, penguji dapat menjelaskan kumpulan kondisi *input* dan melakukan pemeriksaan pada spesifikasi fungsional program.

### 1.2 Pengujian Black Box

Pengujian *Black Box* yaitu pengujian yang membuktikan hasil eksekusi sistem berdasarkan masukan yang diberikan (*test case*) untuk menjamin *fungsional* dari sistem sudah sesuai dengan persyaratan (*requirement*) (Febrian et al., 2020). Pengujian *Black Box* adalah pengujian yang berpusat pada *interface* atau tampilan dan pengujian *fungsional* yang terdapat pada aplikasi serta kesesuaian pada jalannya fungsi yang diperlukan oleh pengguna (Siagian & Lamhot, 2018).

Tahapan dalam Pengujian *Black Box* antara lain (Priyaungga et al., 2020) :

- a. Membuat *test case* untuk menguji fungsi-fungsi yang terletak di aplikasi.
- b. Membuat *test case* untuk menguji ketepatan alur kerja suatu fungsi pada program cocok dengan apa yang dibutuhkan dan permintaan dari pengguna.
- c. Mencari *bugs/error* berdasarkan tampilan (*interface*) pada aplikasi.

Pengujian dengan menggunakan metode *Black Box* biasanya mengarah pada 5 hal, yaitu menemukan fungsi yang tidak benar, kesalahan tampilan antarmuka, menemukan kesalahan pada basis data, menemukan kesalahan pada *performance* sistem, dan terakhir menemukan kesalahan pada *inisialisasi* (Akhirina et al., 2018).

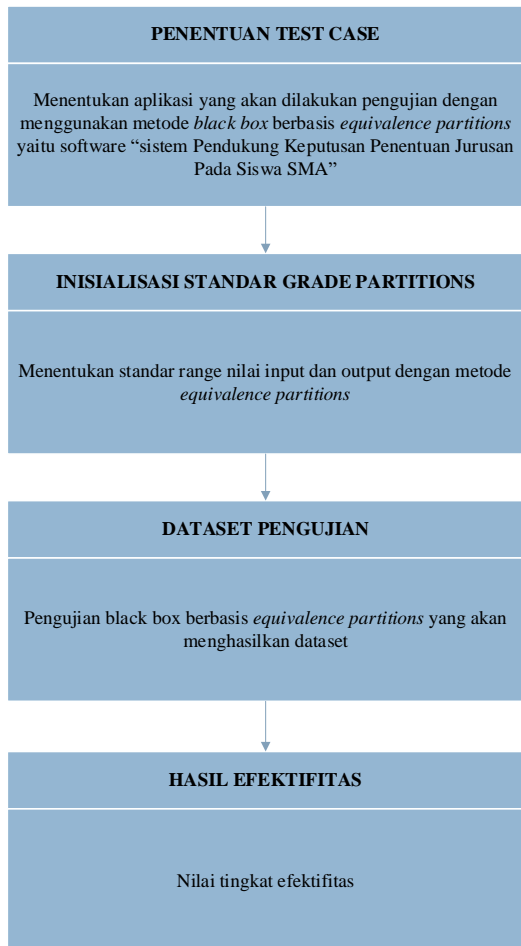
### 1.3 Equivalence Partitions

*Equivalence Partitions* yaitu salah satu teknik dalam *Black Box Testing*. *Equivalence Partitions* yaitu pengujian dimana prosesnya berdasarkan masukkan data pada setiap *form*, teknik pengujian yang memecah domain masukan dari program ke dalam kelas data sehingga *test case* dapat diperoleh. Sehingga, pengujian ini dilakukan dengan memasukkan data yang tidak sesuai dengan *type* data atau memasukkan data acak pada *form* yang ingin diuji (Hanifah & Alit, 2016). Pembuatan *test case Equivalence Partitions* untuk kondisi masukan yang menggambarkan kumpulan keadaan yang sesuai atau tidak (Kesuma Jaya et al., 2019). *Equivalence Partitions* bersumber pada masukan dan keluaran dari suatu komponen yang dipartisi ke dalam kelas-kelas, menurut spesifikasi dari komponen tersebut, yang akan diperlakukan sama (*ekiuvalen*) oleh komponen tersebut (Kesuma Jaya et al., 2019).

## 3. Metodologi

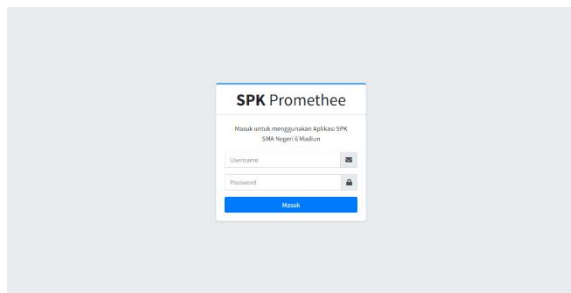
Penelitian ini dilakukan dengan berfokus pada pengujian dengan menggunakan metode *Black Box* berbasis *Equivalence Partitions*. Pada penelitian ini, dilakukan beberapa tahapan antara lain penentuan *test case* aplikasi, *inisialisasi* standar *grade partition input* dan *output*, dataset pengujian dengan metode *Equivalence Partitions* yang akan menghasilkan tingkat efektivitas metode *Equivalence Partitions*.

Pada output pengujian terdapat tabel rancangan *test case* yang digunakan untuk membuat kesimpulan apakah sistem berhasil dalam pengujian atau tidak (Muslimin et al., 2020). Berikut tahapan pengujian *Black Box* berbasis *Equivalence Partitions* dapat dilihat pada Gambar 1 (Rahadi & Vikasari, 2020).



Gambar 1. Tahapan Pengujian *Black Box* Berbasis *Equivalence Partitions*

Pada penelitian ini, fokus pengujian dilakukan pada 3 modul yaitu *fungsional* pada *login* sistem, *fungsional* pada *form* siswa, dan *fungsional* pada *form* nilai siswa. Berikut tampilan dari 3 modul yang telah dijabarkan diatas dapat dilihat pada Gambar 2 hingga Gambar 5.

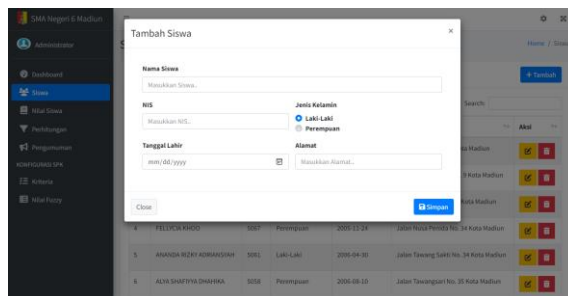


Gambar 2. Tampilan *Form Login*

Berdasarkan *form* pada Gambar 2, terdapat beberapa rancangan pengujian dengan memasukkan data yang tidak *valid* dengan *type* data. Rencana pengujian *username*, data akan sesuai apabila *username* diisi dengan memasukkan angka seperti 0-9 atau huruf seperti a-z ataupun gabungan dari angka atau huruf dengan maksimal 45. Begitupun sebaliknya, data akan tidak sesuai apabila *username* diisi melebihi 45. Rencana pengujian *password*, data akan sesuai apabila *password* diisi dengan memasukkan angka seperti 0-9 atau huruf seperti a-z dengan maksimal 225. Begitupun sebaliknya data akan tidak sesuai apabila *password* diisi melebihi 225. Berikut merupakan hasil dari pengujian *form login* dalam dilihat pada Tabel 1.

Tabel 1. Rancangan *Test Case Form Login*

ID	Deskripsi Pengujian	Hasil yang Diharapkan
L01	Mengisi <i>username</i> dengan "admin" dan <i>password</i> diisi dengan "admin123" dengan syarat data yang diisi adalah data yang terdapat di basis data, kemudian pilih <i>button login</i> .	Sistem berhasil <i>login</i> dan sistem menampilkan halaman <i>dashboard</i> .
L02	Mengisi <i>username</i> dengan "user" dan <i>password</i> diisi dengan "user123" dengan syarat data yang diisi adalah data yang tidak terdapat di basis data, kemudian pilih <i>button login</i> .	Sistem gagal melakukan <i>login</i> dan sistem menampilkan pesan "Username dan Pasword salah!"



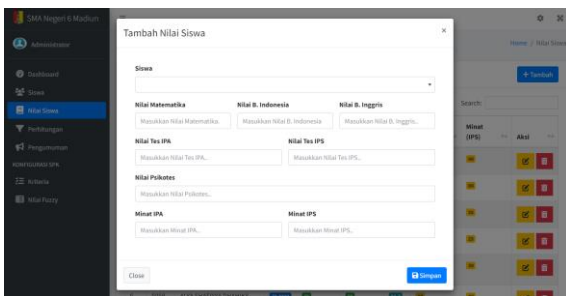
Gambar 3. Tampilan *Form Siswa*

Berdasarkan *form* Gambar 3, terdapat beberapa rancangan pengujian dengan memasukkan data yang tidak *valid* dengan *type* data. Rencana pengujian nama, data akan sesuai apabila nama diisi dengan memasukkan huruf a-z dengan maksimal 100 dan tidak boleh kosong. Begitupun sebaliknya, data akan tidak sesuai apabila nama diisi dengan angka 0-9 serta tidak diisi atau melebihi 100. Rencanan pengujian

NIS, data akan sesuai apabila NIS diisi dengan memasukkan angka 0-9 dengan maksimal 45 dan tidak boleh kosong. Begitupun sebaliknya, data akan tidak sesuai apabila NIS diisi dengan huruf a-z, serta tidak diisi atau melebihi 45. Rencana pengujian alamat, data akan sesuai apabila alamat diisi dengan memasukkan angka 0-9 atau huruf a-z maupun gabungan angka atau huruf dengan maksimal 225 dan tidak boleh kosong. Begitupun sebaliknya, data akan tidak sesuai apabila alamat tidak diisi atau melebihi 255. Berikut merupakan hasil dari pengujian *form* siswa dalam dilihat pada Tabel 2.

Tabel 2. Rancangan *Test Case Form* Siswa

ID	Deskripsi Pengujian	Hasil yang Diharapkan
S01	Memasukkan data nama, NIS, dan alamat pada <i>form</i> siswa dengan benar dan tidak ada yang kosong, kemudian pilih <i>button</i> simpan.	Sistem berhasil menyimpan data pada <i>database</i> dan akan ditampilkan pada tabel siswa.
S02	Memasukkan data nama, NIS, dan alamat pada <i>form</i> siswa dengan acak atau tidak diisi, kemudian pilih <i>button</i> simpan.	Sistem menolak data untuk tersimpan pada <i>database</i> .



Gambar 4 Tampilan *Form* Nilai Siswa

Berdasarkan *form* Gambar 4, terdapat beberapa rancangan pengujian dengan memasukkan data yang tidak *valid* dengan *type* data. Rencana pengujian nilai matematika, nilai Bahasa Indonesia, nilai Bahasa Inggris, nilai tes IPA, nilai tes IPS, nilai psikotes,

minat IPA, dan minat IPS, data akan sesuai apabila keseluruhan nilai yang telah disebutkan di atas diisi dengan memasukkan angka 0-9 dengan maksimal 45 dan tidak boleh kosong. Begitupun sebaliknya, data akan tidak sesuai apabila keseluruhan nilai yang disebutkan di atas diisi dengan huruf a-z, serta tidak diisi atau melebihi 45. Berikut merupakan hasil dari pengujian *form* nilai siswa dalam dilihat pada Tabel 3.

Table 3 Rancangan *Test Case Form* Nilai Siswa

ID	Deskripsi Pengujian	Hasil yang Diharapkan
N01	Memasukkan data nilai matematika, nilai Bahasa Indonesia, nilai Bahasa Inggris, nilai tes IPA, nilai tes IPS, nilai psikotes, minat IPA, dan minat IPS pada <i>form</i> nilai siswa dengan benar yaitu angka seperti 0-9 dan tidak ada yang kosong, kemudian pilih <i>button</i> simpan.	Sistem berhasil menyimpan data pada <i>database</i> dan akan ditampilkan pada tabel nilai siswa.
N02	Memasukkan data nilai matematika, nilai Bahasa Indonesia, nilai Bahasa Inggris, nilai tes IPA, nilai tes IPS, nilai psikotes, minat IPA, dan minat IPS pada <i>form</i> nilai siswa dengan acak atau huruf seperti a-z atau tidak diisi, kemudian pilih <i>button</i> simpan.	Sistem menolak data untuk tersimpan pada <i>database</i> .

#### 4. Hasil dan Pembahasan

Berdasarkan rancangan *test case* yang telah dibuat sebelumnya pada Tabel 1 hingga Tabel 3, maka dapat dilakukan pengujian sistem pendukung keputusan penentuan jurusan pada siswa SMA dengan menggunakan metode *Black Box* berbasis *Equivalence Partitions*. Hasil pengujian dapat dilihat pada Table 4.

Table 4 Hasil pengujian Sistem Pendukung Keputusan Penentuan Jurusan

ID	Deskripsi Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
L01	Mengisi <i>username</i> dengan “admin” dan <i>password</i> diisi dengan “admin123” dengan syarat data yang diisi adalah data yang terdapat di basis data, kemudian pilih <i>button login</i> .	Sistem berhasil <i>login</i> dan sistem menampilkan halaman <i>dashboard</i> .	Sistem masuk ke halaman <i>dashboard</i> .	[ <input checked="" type="checkbox"/> ] Berhasil [ <input type="checkbox"/> ] Gagal
L02	Mengisi <i>username</i> dengan “user” dan <i>password</i> diisi dengan “user123” dengan syarat data yang diisi adalah data yang tidak terdapat di basis data, kemudian pilih <i>button login</i> .	Sistem gagal melakukan <i>login</i> dan sistem menampilkan pesan “Username dan Pasword salah!”	Sistem menolak dan muncul pesan “Username dan Password salah!”	[ <input checked="" type="checkbox"/> ] Berhasil [ <input type="checkbox"/> ] Gagal
S01	Memasukkan data nama, NIS, dan alamat pada <i>form</i> siswa dengan benar dan tidak ada yang kosong, kemudian pilih <i>button simpan</i> .	Sistem berhasil menyimpan data pada <i>database</i> dan akan ditampilkan pada tabel siswa.	Sistem menyimpan data dan terdapat pesan data berhasil disimpan.	[ <input checked="" type="checkbox"/> ] Berhasil [ <input type="checkbox"/> ] Gagal
S02	Memasukkan data nama, NIS, dan alamat pada <i>form</i> siswa dengan acak atau tidak diisi, kemudian pilih <i>button simpan</i> .	Sistem menolak data untuk tersimpan pada <i>database</i> .	Sistem menampilkan pesan kesalahan dan data gagal disimpan.	[ <input checked="" type="checkbox"/> ] Berhasil [ <input type="checkbox"/> ] Gagal
N01	Memasukkan data nilai matematika, nilai Bahasa Indonesia, nilai Bahasa Inggris, nilai tes IPA, nilai tes IPS, nilai psikotes, minat IPA, dan minat IPS pada <i>form</i> nilai siswa dengan benar yaitu angka seperti 0-9 dan tidak ada yang kosong, kemudian pilih <i>button simpan</i> .	Sistem berhasil menyimpan data pada <i>database</i> dan akan ditampilkan pada tabel nilai siswa.	Sistem menyimpan data dan terdapat pesan data berhasil disimpan.	[ <input checked="" type="checkbox"/> ] Berhasil [ <input type="checkbox"/> ] Gagal
N02	Memasukkan data nilai matematika, nilai Bahasa Indonesia, nilai Bahasa Inggris, nilai tes IPA, nilai tes IPS, nilai psikotes, minat IPA, dan minat IPS pada <i>form</i> nilai siswa dengan acak atau huruf seperti a-z atau tidak diisi, kemudian pilih <i>button simpan</i> .	Sistem menolak data untuk tersimpan pada <i>database</i> .	Sistem menampilkan pesan kesalahan dan data gagal disimpan.	[ <input checked="" type="checkbox"/> ] Berhasil [ <input type="checkbox"/> ] Gagal

## 5. Kesimpulan

Pengujian perangkat lunak dengan menggunakan metode *Black Box* bertujuan untuk melihat program dan tugas program tanpa mengetahui *source code* dari program tersebut. *Black Box Testing* berbasis *Equivalence Partitions* dapat membantu pembuatan *test case*, pengujian kualitas, serta dapat menemukan kesalahan yang sebelumnya tidak terdeteksi. Berdasarkan pengujian perangkat lunak sistem pendukung keputusan penentuan jurusan pada siswa SMA dengan menggunakan metode *Black Box* berbasis *Equivalence Partitions* dapat disimpulkan bahwa tidak ditemukan kesalahan pada sistem. Akan

tetapi, diharapkan pada penelitian selanjutnya untuk tidak menggunakan satu metode dalam pengujian agar hasil yang didapat bisa lebih valid lagi.

**Daftar Pustaka:**

- Adi, R. P., Koswara, Y., Tashika, J., Devi, Y., & Saifudin, A. (2020). Pengujian Black Box pada Aplikasi Pertokoan Minimarket Menggunakan Metode Equivalence Partitioning. *Jurnal Teknologi Sistem Informasi Dan Aplikasi*, 3(2), 100. <https://doi.org/10.32493/jtsi.v3i2.4695>
- Akhirina, T. Y., Yulistiyanti, D., Rusmardiana, A., & Pauziah, U. (2018). Pengujian Sistem Pendukung Keputusan Jurusan SMA di Banten menggunakan Metode Black Box. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 2(3), 800–806. <https://doi.org/10.29207/resti.v2i3.615>
- Debiyanti, D., Sutrisna, S., Budrio, B., Kamal, A. K., & Yulianti, Y. (2020). Pengujian Black Box pada Perangkat Lunak Sistem Penilaian Mahasiswa Menggunakan Teknik Boundary Value Analysis. *Jurnal Informatika Universitas Pamulang*, 5(2), 162. <https://doi.org/10.32493/informatika.v5i2.5446>
- Febrian, V., Ramadhan, M. R., Faisal, M., & Saifudin, A. (2020). Pengujian pada Aplikasi Penggajian Pegawai dengan menggunakan Metode Blackbox. *Jurnal Informatika Universitas Pamulang*, 5(1), 61. <https://doi.org/10.32493/informatika.v5i1.4340>
- Hanifah, U., & Alit, R. (2016). *Penggunaan metode black box pada pengujian sistem informasi surat keluar masuk. XI*.
- Hidayat, T., & Muttaqin, M. (2018). Pengujian Sistem Informasi Pendaftaran dan Pembayaran Wisuda Online menggunakan Black Box Testing dengan Metode Equivalence Partitioning dan Boundary Value Analysis. *Jurnal Teknik Informatika UNIS JUTIS*, 6(1), 2252–5351. [www.ccsenet.org/cis](http://www.ccsenet.org/cis)
- Kesuma Jaya, M. S. A., Gumilang, P., Wati, T., Andersen, Y. P., & Desyani, T. (2019). Pengujian Black Box pada Aplikasi Sistem Penunjang Keputusan Seleksi Calon Pegawai Negeri Sipil Menggunakan Teknik Equivalence Partitions. *Jurnal Informatika Universitas Pamulang*, 4(4), 131. <https://doi.org/10.32493/informatika.v4i4.3834>
- 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*, 3(1), 50. <https://doi.org/10.32493/jtsi.v3i1.4307>
- Muslimin, D. B., Kusmanto, D., Amilia, K. F., Ariffin, M. S., Mardiana, S., & Yulianti, Y. (2020). Pengujian Black Box pada Aplikasi Sistem Informasi Akademik Menggunakan Teknik Equivalence Partitioning. *Jurnal Informatika Universitas Pamulang*, 5(1), 19. <https://doi.org/10.32493/informatika.v5i1.3778>
- 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*, 4(4), 125. <https://doi.org/10.32493/informatika.v4i4.3782>
- Priyaangga, 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>
- Rahadi, N. W., & Vikasari, C. (2020). Pengujian Software Aplikasi Perawatan Barang Milik Negara Menggunakan Metode Black Box Testing Equivalence Partitions. *Infotekmesin*, 11(1), 57–61. <https://doi.org/10.35970/infotekmesin.v11i1.124>
- Snadhika Jaya, T. (2018). Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung). *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, 03(02), 45–48.
- Trengginaz, R. B., Yusup, A., Sunyoto, D. S., Jihad, M. R., & Yulianti, Y. (2020). Pengujian Aplikasi Pemesanan Tiket Kereta berbasis Website Menggunakan Metode Black Box dengan Teknik Equivalence Partitioning. *Jurnal Teknologi Sistem Informasi Dan Aplikasi*, 3(3), 144. <https://doi.org/10.32493/jtsi.v3i3.5349>
- Utomo, A., Sutanto, Y., Tiningrum, E., & Susilowati, E. M. (2020). Pengujian Aplikasi Transaksi Perdagangan Menggunakan Black Box Testing Boundary Value Analysis. *Jurnal Bisnis Terapan*, 4(2), 133–140. <https://doi.org/10.24123/jbt.v4i2.2170>