

Tinjauan Langkah Pengujian Perangkat Lunak Dengan Metode *Black Box* Menggunakan Teknik *Equivalence Partitions*

Abi Mayu Abdi

Sistem Informasi, Sains dan Teknologi, UIN Sulthan Thaha Saifuddin Jambi

*e-mail: abimayuabdi@gmail.com

Abstrak

Pengujian perangkat lunak merupakan tahap mengidentifikasi kebenaran perangkat lunak sebelum di publikasikan kepada pengguna. Dengan memastikan bahwa komponen tersebut memenuhi persyaratan yang ditentukan dan prosesnya juga memberikan informasi kepada klien tentang kualitas perangkat lunak. *Black Box testing* adalah salah satu metode pengujian perangkat lunak dengan fokus pengujian pada bagian fungsionalitas perangkat lunak. Teknik pengujian Black Box yang digunakan dalam pembahasan ini adalah teknik *Partitions Equivalence*. Teknik *Partitions Equivalence*, pengujian berdasarkan masukkan data pada setiap form yang ada, setiap menu masukan akan dilakukan pengujian dan dikelompokkan berdasarkan fungsinya baik itu bernilai valid ataupun tidak valid. Dengan teknik ini pengujian dapat fokus pada mewakili setiap partisi dengan satu atau beberapa nilai input yang mewakili keseluruhan partisi tersebut. Tujuan utama dari teknik ini adalah untuk mengurangi jumlah uji yang diperlukan untuk mencakup seluruh domain input, sambil tetap memastikan bahwa setiap partisi telah diuji dengan benar. Penggunaan teknik *Partitions Equivalence* membantu menyederhanakan pengujian dan membuatnya lebih terarah, sehingga waktu dan sumber daya yang diperlukan untuk pengujian dapat dikelola dengan lebih efisien.

Kata kunci: Pengujian, Perangkat lunak, *Black box*, dan *Partitions Equivalence*

Abstract

Software testing is the stage of identifying the correctness of the software before it is published to users. By ensuring that the components meet the specified requirements and the process also provides the client with information about the quality of the software. Black Box testing is a software testing method with a testing focus on the functionality of the software. The Black Box testing technique used in this discussion is the Partitions Equivalence technique. Partitions Equivalence technique, testing based on entering data in each existing form, each input menu will be tested and grouped based on its function, whether it is valid or invalid. With this technique testing can focus on representing each partition with one or more input values that represent the entire partition. The main goal of this technique is to reduce the number of tests required to cover the entire input domain, while still ensuring that each partition is properly tested. Using the Partitions Equivalence technique helps simplify testing and make it more targeted, so that the time and resources required for testing can be managed more efficiently.

Keywords: Testing, Software, Black box, and Partitions Equivalence

1 Pendahuluan (or Introduction)

Pengujian perangkat lunak merupakan tahapan mengidentifikasi kebenaran perangkat lunak dengan mempertimbangkan semua atributnya seperti keandalan, skalabilitas, portabilitas, kegunaan dan mengevaluasi eksekusi komponen perangkat lunak untuk menemukan bug atau cacat perangkat lunak. Pengujian black box menguji fungsionalitas perangkat lunak tanpa melihat struktur internal atau pengkodeannya. Sumber utama pengujian black box adalah spesifikasi persyaratan yang dinyatakan oleh pelanggan.[1]

Pengujian pada sebuah program penting untuk dilakukan menghindari kerugian yang akan ditimbulkan dari kesalahan perangkat lunak [2]. Pengujian perangkat lunak memberikan pandangan dan tujuan independen dari perangkat lunak sehingga memberikan jaminan kesesuaian perangkat lunak. Dan memastikan bahwa komponen tersebut memenuhi persyaratan yang ditentukan. Prosesnya juga memberikan informasi kepada klien tentang kualitas perangkat lunak.

Jenis pengujian perangkat lunak dapat dikelompokkan menjadi dua, yaitu pengujian otomatis dan pengujian manual. Pada pengujian manual, seorang tester tidak menggunakan alat bantu pengujian dan tidak memerlukan pengetahuan khusus tentang alat pengujian. Melainkan membutuhkan pemahaman yang baik tentang produk perangkat lunak sehingga dapat menyiapkan dokumen pengujian. Metode pengujian manual terdiri dari *white box testing*, *black box testing* dan *gray box testing* [3].

Metode pengujian *white box testing*, *black box testing* dan *gray box testing* dilakukan untuk saling melengkapi dalam tahap pengujian perangkat lunak. Seperti pengujian *black box testing* digunakan untuk melengkapi pengujian *white box testing*, supaya menghasilkan perangkat lunak yang berkualitas [4]. Pengujian *black box testing* merupakan pengujian yang berfokus kepada fungsionalitas perangkat lunak dengan mengidentifikasi masukan lalu diuji agar kita mengetahui letak kesalahannya [5]. *Black box testing* dilakukan oleh *test engineer* dengan memeriksa fungsionalitas perangkat lunak sesuai dengan kebutuhan pelanggan. Selaras dengan namanya pengujian kotak hitam, dalam hal ini kode tidak terlihat saat melakukan pengujian.

Terdapat beberapa teknik dalam metode pengujian *black box testing*, salah satunya adalah teknik *Partitions Equivalence*. Teknik ini, melakukan pengujian berdasarkan masukan data pada setiap form yang ada, setiap menu masukan akan dilakukan pengujian dan dikelompokkan berdasarkan fungsinya baik itu bernilai valid ataupun tidak valid [4]. Apabila suatu kondisi suatu partisi benar, maka kondisi partisi lain yang setara juga harus benar. Dan apabila kondisi suatu partisi salah, maka kondisi partisi lain yang setara juga harus salah.

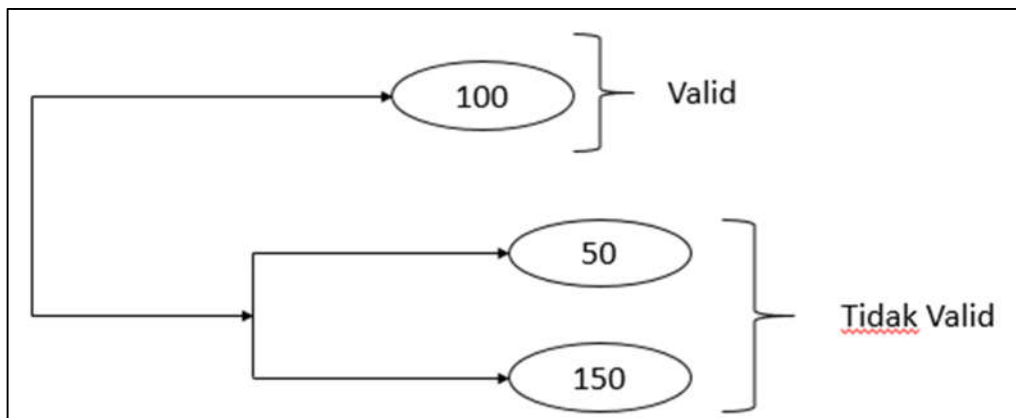
Dengan teknik *Partitions Equivalence* pengujian dapat fokus pada mewakili setiap partisi dengan satu atau beberapa nilai input yang mewakili keseluruhan partisi tersebut. Tujuan utama dari teknik ini adalah untuk mengurangi jumlah uji yang diperlukan untuk mencakup seluruh domain input, sambil tetap memastikan bahwa setiap partisi telah diuji dengan benar. Penggunaan teknik *Partitions Equivalence* membantu menyederhanakan pengujian dan membuatnya lebih terarah, sehingga waktu dan sumber daya yang diperlukan untuk pengujian dapat dikelola dengan lebih efisien.

2 Tinjauan Literatur

Partitions Equivalence adalah teknik pengujian perangkat lunak di mana data masukan dibagi menjadi partisi dengan nilai yang valid dan tidak valid, dan semua partisi harus menunjukkan perilaku yang sama. Jika suatu kondisi suatu partisi benar, maka kondisi partisi lain yang setara juga harus benar, dan jika kondisi suatu partisi salah, maka kondisi partisi lain yang setara juga harus salah. Prinsip partisi ekuivalensi adalah, kasus uji harus dirancang untuk mencakup setiap partisi setidaknya satu kali. Setiap nilai dari setiap partisi yang sama harus menunjukkan perilaku yang sama seperti yang lain.

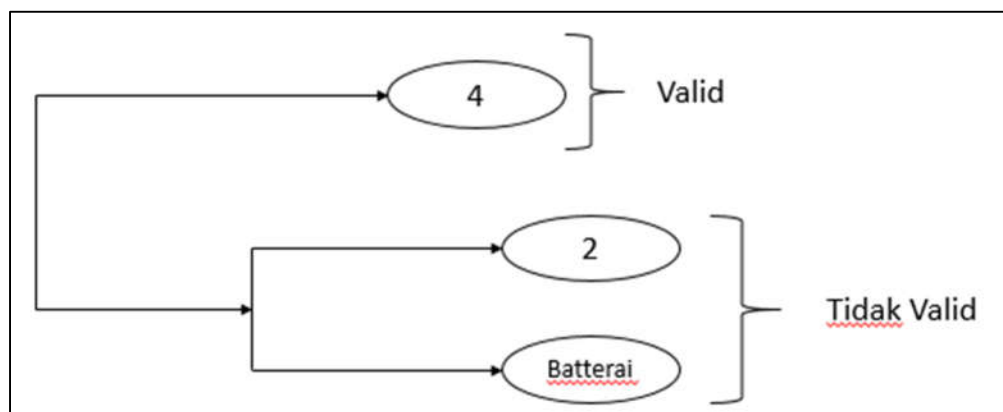
Partitions Equivalence berasal dari persyaratan dan spesifikasi perangkat lunak. Keuntungan dari pendekatan ini adalah membantu mengurangi waktu pengujian karena jumlah kasus pengujian yang lebih sedikit dari tak terbatas menjadi terbatas. Ini berlaku di semua tingkat proses pengujian. Dalam melakukan *Partitions Equivalence* dilakukan dengan pendekatan Pressman. Berikut bagaimana pendekatan pressman dalam beberapa kondisi [6]:

- a. Jika kondisi input adalah rentang nilai, maka kasus uji untuk satu masukan yang valid dan dua masukan yang tidak valid. Contoh, nilai ujian kelayakan pengetahuan teori pengembangan perangkat lunak mahasiswa dikatakan layak adalah dengan rentang (80-120).



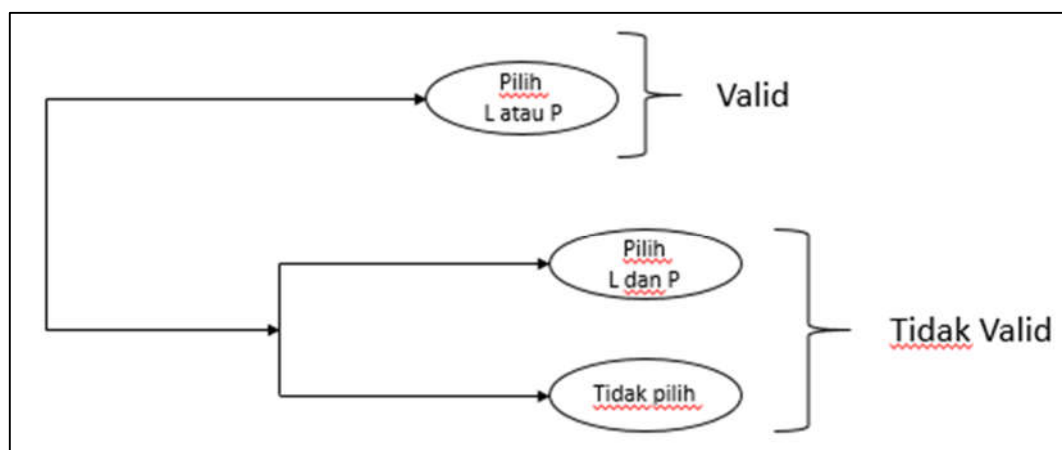
Gambar 1 Kondisi Input adalah rentang nilai

- b. Jika kondisi input adalah sekumpulan nilai, maka kasus uji untuk satu masukan yang valid dan dua masukan yang tidak valid. Contoh, melakukan belanja produk komputer dengan dengan beberapa id produk yaitu 1, 4, 7 dan 9. Dengan keterangan id 1 adalah mouse, id 4 adalah pelindung layar, id 7 adalah charger, dan id 9 adalah pelindung keyboard.



Gambar 2 Kondisi Input adalah kumpulan nilai

- c. Jika kondisi input adalah *boolean* (benar atau salah), maka kasus uji untuk kedua nilai benar atau salah. Contoh, memilih jenis kelamin L atau P dengan *radio button* pada sebuah form.

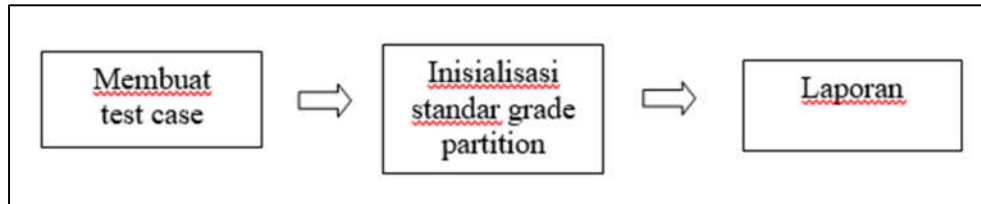


Gambar 3 Kondisi Input adalah Boolean

3 Metode Penelitian

Tahapan metode Black Box Testing dengan teknik *Partitions Equivalence* adalah,

- Membuat *Test Case* perangkat lunak yang akan diuji dengan metode *Equivalence Partitions*
- Menginisialisasi *standar grade partition* masukan dan keluaran
- Laporan hasil pengujian



Gambar 4 Tahapan pengujian

4 Hasil dan Pembahasan

- Membuat *Test Case* perangkat lunak yang akan diuji dengan metode *Equivalence Partitions*

Gambar 5 Form Input OTP

Asumsikan ada sebuah fungsi aplikasi perangkat lunak yang menerima sejumlah digit tertentu, tidak lebih besar atau lebih kecil dari angka tersebut. Misalnya nomor OTP yang hanya berisi 6 digit. Apabila kurang atau lebih dari enam digit, aplikasi tidak akan diterima, dan akan mengarahkan pengguna ke halaman kesalahan.

Tabel 1 Rancangan Test Case Form Input OTP

Id	Deskripsi pengujian	Hasil yang diharapkan
O01	Digit ≥ 7 , misalkan 1234567	<ul style="list-style-type: none"> Valid, aplikasi akan menerima dan mengarahkan pengguna ke halaman selanjutnya Invalid, aplikasi akan mengarahkan pengguna ke halaman kesalahan
O02	Digit ≤ 5 , misalkan 54321	<ul style="list-style-type: none"> Valid, aplikasi akan menerima dan mengarahkan pengguna ke halaman selanjutnya Invalid, aplikasi akan mengarahkan pengguna ke halaman kesalahan
O03	Digit $= 6$, misalkan 234567	<ul style="list-style-type: none"> Valid, aplikasi akan menerima dan mengarahkan pengguna ke halaman selanjutnya Invalid, aplikasi akan mengarahkan pengguna ke halaman kesalahan

Contoh berikut nya, asumsikan ada sebuah fungsi aplikasi perangkat lunak yang menerima 10 digit nomor ponsel di gambar 6. Apabila kurang atau lebih dari 10 digit, aplikasi akan menolak, dan akan mengarahkan pengguna ke halaman kesalahan.

Gambar 6 Form Input Nomor Ponsel

Dari gambar 6 dapat disusun test case pengujian sebagai skenario pengujian, berikut test case di tabel 2.

Tabel 2 Rancangan Test Case Form Input OTP

Id	Deskripsi pengujian	Hasil yang diharapkan
N01	Digit ≥ 11 , misalkan 082170881180	<ul style="list-style-type: none"> Valid, aplikasi akan menerima dan mengarahkan pengguna ke halaman selanjutnya Invalid, aplikasi akan mengarahkan pengguna ke halaman kesalahan
N02	Digit ≤ 9 , misalkan 082170881	<ul style="list-style-type: none"> Valid, aplikasi akan menerima dan mengarahkan pengguna ke halaman selanjutnya Invalid, aplikasi akan mengarahkan pengguna ke halaman kesalahan
N03	Digit =10, misalkan 0821082111	<ul style="list-style-type: none"> Valid, aplikasi akan menerima dan mengarahkan pengguna ke halaman selanjutnya Invalid, aplikasi akan mengarahkan pengguna ke halaman kesalahan

- b. Menginisialisasi *standar grade partition* masukan dan keluaran

Dari situasi yang ada , dilakukan inisialisasi *standar grade partition* sebagai dasar dalam melakukan pengujian. Berikut *standar grade partition* form input OTP dan input nomor ponsel di tabel 3.

Tabel 3 Standar grade partition

Fungsi aplikasi	Standar masukan	Standar keluaran
Input OTP	6 digit	Halaman selanjutnya
Input nomor ponsel	10 digit	Halaman selanjutnya

c. Laporan hasil pengujian

Berdasarkan rancangan uji kasus (test case) yang telah dibuat pada Tabel 1 dan 2, maka dapat dilakukan pengujian. Hasil pengujian ditunjukkan pada Tabel 4.

Tabel 4 Hasil pengujian

Id	Deskripsi pengujian	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
O01	Digit ≥ 7 , misalkan 1234567	Invalid, aplikasi akan mengarahkan pengguna ke halaman kesalahan	Muncul notifikasi, inputan salah	Sesuai
O02	Digit ≤ 5 , misalkan 54321	Invalid, aplikasi akan mengarahkan pengguna ke halaman kesalahan	Muncul notifikasi, inputan salah	Sesuai
O03	Digit =6, misalkan 234567	Valid, Valid, aplikasi akan menerima dan mengarahkan pengguna ke halaman selanjutnya	Muncul notifikasi, data berhasil disimpan	Sesuai
N01	Digit ≥ 11 , misalkan 082170881180	Invalid, aplikasi akan mengarahkan pengguna ke halaman kesalahan	Muncul notifikasi, inputan salah	Sesuai
N02	Digit ≤ 9 , misalkan 082170881	Invalid, aplikasi akan mengarahkan pengguna ke halaman kesalahan	Muncul notifikasi, inputan salah	Sesuai
N03	Digit =10, misalkan 0821082111	Valid, Valid, aplikasi akan menerima dan mengarahkan pengguna ke halaman selanjutnya	Muncul notifikasi, data berhasil disimpan	Sesuai

5 Kesimpulan

Pengujian *black box* adalah salah satu teknik dalam pengujian perangkat lunak yang berfokus pada pengujian fungsionalitas suatu sistem berdasarkan spesifikasi yang diberikan, tanpa memeriksa kode sumber atau struktur internal perangkat lunak. Teknik ini sangat bermanfaat untuk memvalidasi apakah perangkat lunak memenuhi persyaratan pengguna dan berfungsi sesuai dengan tujuan yang diinginkan. Pengujian *black box* dapat digunakan pada berbagai tingkatan pengujian, seperti pengujian unit, integrasi, sistem, dan penerimaan, dengan sumber utama yang berasal dari dokumen spesifikasi atau kebutuhan pengguna.

Salah satu pendekatan yang sering digunakan dalam pengujian *black box* adalah *Equivalence Partitioning* atau *Partition Equivalence*. Teknik ini membagi data input ke dalam beberapa kelompok atau partisi yang dianggap ekuivalen. Setiap partisi mewakili serangkaian nilai input yang diharapkan menghasilkan perilaku yang sama dari perangkat lunak. Dengan menguji hanya satu nilai dari setiap partisi, pengujian dapat mengurangi jumlah pengujian yang diperlukan, sehingga waktu pelaksanaan pengujian menjadi lebih efisien tanpa mengorbankan cakupan pengujian. Selain itu, teknik ini memungkinkan pengujian untuk mengidentifikasi kesalahan dalam berbagai kategori input secara efektif.

Namun, kelemahan utama dari teknik *Equivalence Partitioning* adalah ketidakmampuannya untuk menangani kasus-kasus ekstrem atau nilai-nilai batas (*boundary values*) secara eksplisit. Nilai-nilai batas, seperti nilai minimum, maksimum, atau nilai yang berada di sekitar batas validitas data, sering kali menjadi sumber utama kesalahan perangkat lunak. Oleh karena itu, untuk mengatasi kekurangan ini, disarankan untuk menggunakan teknik lain seperti *Boundary Value Analysis* (BVA). Teknik BVA dirancang khusus untuk memeriksa perilaku perangkat lunak pada nilai-nilai batas ini, sehingga memberikan cakupan pengujian yang lebih komprehensif. Kombinasi antara *Equivalence Partitioning*

dan *Boundary Value Analysis* dapat memberikan pendekatan pengujian yang lebih efektif dan menyeluruh.

Dengan demikian, pengujian *black box* menawarkan metode yang kuat untuk memvalidasi fungsi perangkat lunak, terutama jika dilakukan dengan memanfaatkan teknik-teknik seperti *Equivalence Partitioning* dan *Boundary Value Analysis*. Kombinasi ini memungkinkan pengujian yang efisien sekaligus memastikan bahwa perangkat lunak dapat menangani skenario input dengan baik, termasuk kasus normal dan kasus ekstrem.

Referensi

- [1] Y. Ramadhani, "Pengujian Black Box," <http://www.dadosen.com/2023/11/pengujian-black-box.html>.
- [2] A. Sethi, "a Review Paper on Levels, Types& Techniques in Software Testing," *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 7, pp. 269–271, 2017, doi: 10.26483/ijarcs.v8i7.4236.
- [3] Javatpoint.com, "Black box testing," <https://www.javatpoint.com/black-box-testing>.
- [4] M. S. Mustaqbal, R. F. Firdaus, and H. Rahmadi, "PENGUJIAN APLIKASI MENGGUNAKAN BLACK BOX TESTING BOUNDARY VALUE ANALYSIS (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN)," vol. I, no. 3, pp. 31–36, 2015.
- [5] A. A. Arwaz, T. Kusumawijaya, R. Putra, K. Putra, and A. Saifudin, "Pengujian Black Box pada Aplikasi Sistem Seleksi Pemenang Tender Menggunakan Teknik Equivalence Partitions," *J. Teknol. Sist. Inf. dan Apl.*, vol. 2, no. 4, p. 130, 2019, doi: 10.32493/jtsi.v2i4.3708.
- [6] R. S. Pressman, *Rekayasa Perangkat Lunak. Pendekatan Praktisi. Edisi 7*. 2012.