

Modifikasi pada Pembangkit Kasus Uji untuk Pengujian Kombinasi

Modification on Test Case Generator for Pairwise Testing

Hidayatulhidayah¹, Sri Widowati², Donni Richasdy³

^{1,2,3} Universitas Telkom, Bandung

¹hidayatulhidayah@student.telkomuniversity.ac.id, ²sriwidowati@telkomuniversity.ac.id,

³donnir@telkomuniversity.ac.id

Abstrak

Pairwise testing adalah salah satu teknik pengujian black box yang berdasarkan pada input parameter. Pairwise testing, atau dikenal dengan All-pairs testing menerapkan kombinatorial dalam membangun kasus uji. Yaitu dengan melakukan kombinasi dari setiap parameter input untuk membentuk data set pada kasus uji. Untuk mendapatkan kasus uji dengan metode ini, terdapat pembangkit kasus uji yang menerapkan metode pairwise testing. Namun, pada pembangkit tersebut memiliki kekurangan dalam fungsi yang dapat dipilih. Setelah melakukan analisis terhadap pembangkit kasus uji, ditemukan bahwa pembangkit tidak memiliki fungsi filter untuk membuat data set. Fungsi filter berfungsi untuk memberikan batasan tertentu pada data input sebelum pembangkit mengubah data input menjadi set kasus uji. Penelitian ini bertujuan untuk memodifikasi sebuah pembangkit kasus uji dengan menambahkan fungsi filter. Testcase yang dihasilkan oleh pembangkit setelah menggunakan fungsi filter telah mampu mengikuti batasan atau filtering yang diberikan pada data input.

Kata kunci: Filter, Kombinasi, Pairwise testing, Pengujian

Abstract

Pairwise testing is a black box testing technique based on input parameters. Pairwise testing, also known as All-pairs testing, applies combinatorial testing in building test cases. That is combined of each input parameter to form a data set in the test case. To get test cases with this method, there is a test case generator that applies the pairwise testing method. However, these generators have shortcomings in the functions that can be selected. After analyzing the test case generator, it was found that the generator does not have a filter function to create a data set. The filter function serves to provide certain constraints on the input data before the generator converts the input data into a set of test cases. This study aims to modify a test case generator by adding a filter function. The testcase generated by the generator after using the filter function has been able to follow the limits or filtering given to the input data.

Keywords: Filter, Combination, Pairwise testing, Testing

1. Pendahuluan

Latar Belakang

Pengujian perangkat lunak menjadi hal yang mendasar selama masa pengembangan perangkat lunak. Terdapat dua jenis pengujian perangkat lunak, yaitu *white-box testing* yang memeriksa kode program secara detail pada tingkat logika internal, dan *black-box testing* yang menguji perangkat lunak pada tingkat aspek dasar sistem [1]. Salah satu pengujian *black-box testing*, berdasarkan teknik pengujian input domain adalah *pairwise testing* [2]. *Pairwise testing* adalah teknik pengujian kombinatorial yang berupaya memastikan bahwa, kasus uji berfokus pada pendefinisian set tes yang menyediakan tes pasangan variabel, bukan tes yang menggabungkan semua variabel [3]. Artinya, terdapat pengurangan jumlah kasus uji dalam penerapan *pairwise testing*. Pairwise testing menerapkan teknik kombinasi yang di eksplorasi dengan tujuan menggabungkan nilai input perangkat lunak secara otomatis berdasarkan strategi tertentu. Yaitu dengan memvariasikan nilai dari setiap pasangan parameter input (data set). Sehingga semua kemungkinan kombinasi dari parameter input tersebut telah dibuat. Lalu, metode pairwise testing ini menerapkan

pendekatan Orthogonal Array, yaitu pendekatan sistematis yang membantu pairwise testing untuk tidak membuat keseluruhan hasil kombinasi dari parameter input. Teknik pairwise testing terbukti dalam beberapa studi bahwa relatif lebih baik dalam mendeteksi kesalahan perangkat lunak [4]. Pada sebuah penelitian menyimpulkan bahwa pengujian kombinasi sangat berguna dalam cakupan kode program dan penemuan kesalahan pada program. Bahkan, pengujian kombinatorial sama efektifnya dengan pengujian manual sebanyak 64% dalam menemukan kesalahan program. Pairwise testing merupakan teknik kombinatorial yang paling banyak digunakan, karena merupakan strategi yang efektif diterapkan pada pengujian perangkat lunak [5]. Pairwise testing dapat menemukan sebagian besar kesalahan dan berfokus menghasilkan sebuah set data test case dari semua input parameter menjadi test case dengan jumlah yang sedikit.

Membuat rangkaian kasus uji atau *test case* dapat melalui beberapa cara, secara garis besar adalah dengan menggunakan pembangkit kasus uji atau dengan analisis manual. Kasus uji yang dihasilkan melalui pembangkit ini dapat memiliki hasil yang berbeda-beda, sesuai dengan metode yang digunakan untuk membangun pembangkit tersebut. Salah satunya adalah dengan pembangkit kasus uji yang menggunakan metode kombinasi atau Pairwise Testing. Telah banyak dilakukan penelitian mengenai pembangkit kasus uji dengan metode pairwise testing, bahkan terdapat generator online untuk menghasilkan data set dengan metode kombinasi ini. Kemudian, mengacu pada penelitian sebelumnya mengenai kakas pairwise testing sebagai referensi utama [6], dilakukan analisis pada pembangkit kasus uji dengan metode pairwise testing. Pembangkit kasus uji dengan metode pairwise testing dapat menghasilkan kasus uji yang sesuai dengan prinsip kerja pairwise testing. Namun, terdapat pembangkit kasus uji dengan metode ini yang memiliki beberapa kekurangan. Salah satunya adalah fungsi untuk memberikan filtering pada data set. Contoh kecilnya adalah, komputer tipe A,B dan C akan memproses data X,Y dan Z. Secara utuh dengan menerapkan pairwise testing pada data tersebut, akan dibangun kasus uji 3x3 yaitu 9 kasus uji. Namun apabila parameter memiliki syarat yaitu komputer tipe A tidak dapat memproses data X maka kasus uji akan berkurang menjadi 8 kasus uji.

Oleh karena itu, diperlukan suatu modifikasi pada pembangkit kasus uji dengan metode pairwise testing, untuk melengkapi kekurangan pada fungsi yang tidak ada pada pembangkit, dalam hal ini adalah fungsi filter. Penelitian ini bertujuan untuk menambahkan suatu fungsi pada pembangkit kasus uji dengan metode pairwise testing sebagai modifikasi minor pada pembangkit. Agar pembangkit kasus uji dapat melakukan filtering atau saringan terhadap data input untuk membuat kasus uji.

Rumusan Masalah

Adapun pertanyaan pada penelitian atau Reserch Question (RQ) untuk memecahkan masalah dalam proses menghasilkan testcase dengan fungsi filter pada pairwise testing yaitu:

1. Apakah fungsi filter yang ditambahkan pada generator kasus uji sudah mampu melakukan filtering terhadap testcase?
2. Bagaimana perbedaan hasil testcase dengan filtering dan tanpa filtering?
3. Bagaimana proses eliminasi set data yang yang dihasilkan oleh fungsi filtering?

Tujuan

Menjawab rumusan masalah diatas, berikut beberapa tujuan yang akan dicapai dalam penelitian:

1. Menambahkan fungsi filter pada *generator* kasus uji untuk pairwise testing agar mampu melakukan filtering terhadap testcase.
2. Melakukan perbandingan hasil testcase dengan filtering dan tanpa filtering.
3. Menjelaskan proses eliminasi set data yang dihasilkan oleh fungsi filtering dari kombinasi utuh data uji.

Batasan Masalah

Adapun batasan masalah pada penelitian ini pertama, memiliki keterbatasan orang untuk melakukan pengujian dalam menghasilkan testcase menggunakan fungsi ini. Kedua, tidak dapat menerapkan data yang besar karena tidak memiliki izin penggunaan studi kasus dengan entri yang banyak. Lalu, metode pairwise testing ini dapat memberikan dampak yang kuat dalam pengurangan jumlah *testcase* apabila digunakan pada data pengujian yang besar dan konfigurasi yang banyak.

Organisasi Tulisan

Bagian 2 berisikan studi literatur yang terkait dengan penelitian. Studi literatur merupakan teori yang dapat dijadikan acuan pada penelitian Pairwise testing. Bagian 3 berisikan sistem yang dibangun beserta skemanya, pada skema atau proses yang dibuat menjelaskan bagaimana melakukan modifikasi dan hasil yang diberikan.

2. Studi Terkait

2.1. Pairwise Testing

Dalam hal pengujian, terdapat sebuah studi terkini yang berfokus pada pembangkitan atau pengecilan jumlah kasus uji berdasarkan kekuatan interaksi yang disebut dengan *Combinatorial Interaction Testing* (CIT). Sebuah sistem perangkat lunak yang bersifat *customizable* (contoh: Microsoft Word), dapat memiliki sejumlah pilihan konfigurasi yang banyak, yang butuh untuk diuji. CIT adalah contoh teknik yang sering digunakan untuk menguji konfigurasi pada sistem sejenis dengan setiap *t-combination* (t adalah kekuatan interaksi) dari nilai inputan, telah terpenuhi pada kasus uji setidaknya satu kali. Kasus khusus pada CIT adalah *pairwise testing*. Pairwise testing adalah teknik yang efektif, berdasarkan *failures* yang sering terjadi pada perangkat lunak disebabkan karena interaksi antara dua parameter. Dari CIT yang sudah ada, keefektifan dari pairwise testing telah dievaluasi pada beberapa penelitian. Studi yang dilakukan oleh Kuhn, Wallace terhadap beberapa sistem menyatakan bahwa pairwise testing sangat efektif untuk mendeteksi kesalahan interaksi. Studi Search based Software Engineering pada penerapan pairwise testing menemukan bahwa 76% dari *bugs* dapat dideteksi dengan pairwise testing. Kemudian sebuah studi oleh Brownlie, Prowse, and Phadke menunjukkan bahwa menerapkan pairwise testing dapat mengurangi kasus uji dari 1000 kasus menjadi 422 kasus [7].

Hal ini menunjukkan pengurangan jumlah kasus uji dengan penerapan pairwise testing sangat *significant* hingga mencapai kurang dari setengah jumlah kasus uji awal. Pengurangan jumlah kasus uji ini adalah berdasarkan teknik yang didesain untuk dapat membangun pasangan tes dengan usaha yang sedikit, yaitu *Combinatorial testing*. *Combinatorial testing* adalah teknik yang membuat test input dengan melakukan kombinasi antara nilai masukan. Kemudian, pairwise testing adalah teknik yang menggunakan pendekatan menuju combinatorial testing dengan melakukan *generate* pada pasangan tes yang mencakup setiap hasil kombinasi.

2.2. Orthogonal Array

Salah satu pendekatan matematis yang diterapkan dalam membangun kasus uji pada metode pairwise testing adalah Orthogonal Array. Orthogonal Array dapat membantu untuk memaksimalkan capaian kasus uji dengan meminimalkan jumlah kasus uji. Orthogonal Array adalah skema klasifikasi dua arah yang seimbang, untuk membangun eksperimen yang lebih praktis dengan tidak membangun kasus uji dari seluruh hasil kombinasi parameter input [8]. Ukuran dan bentuk array yang dihasilkan tergantung dari parameter input untuk membentuk array. Orthogonal Array di simbolkan sebagai berikut:

$$L_4(2^3)$$

Dengan angka 4 adalah jumlah baris, 2 adalah jumlah nilai maksimum input dan 3 adalah jumlah kolom array. Metode yang digunakan dalam Orthogonal Array merupakan hasil eksperimental dari konstruksi literatur statistik. Oleh karena itu, membangun array dengan Orthogonal array bergantung dari konstruksi kombinatorial. Dari rumusan di atas, array yang terbentuk adalah:

Tabel 1 Orthogonal Array $L_4(2^3)$

	1	2	3
1	1	1	1
2	1	2	2
3	2	1	2
4	2	2	1

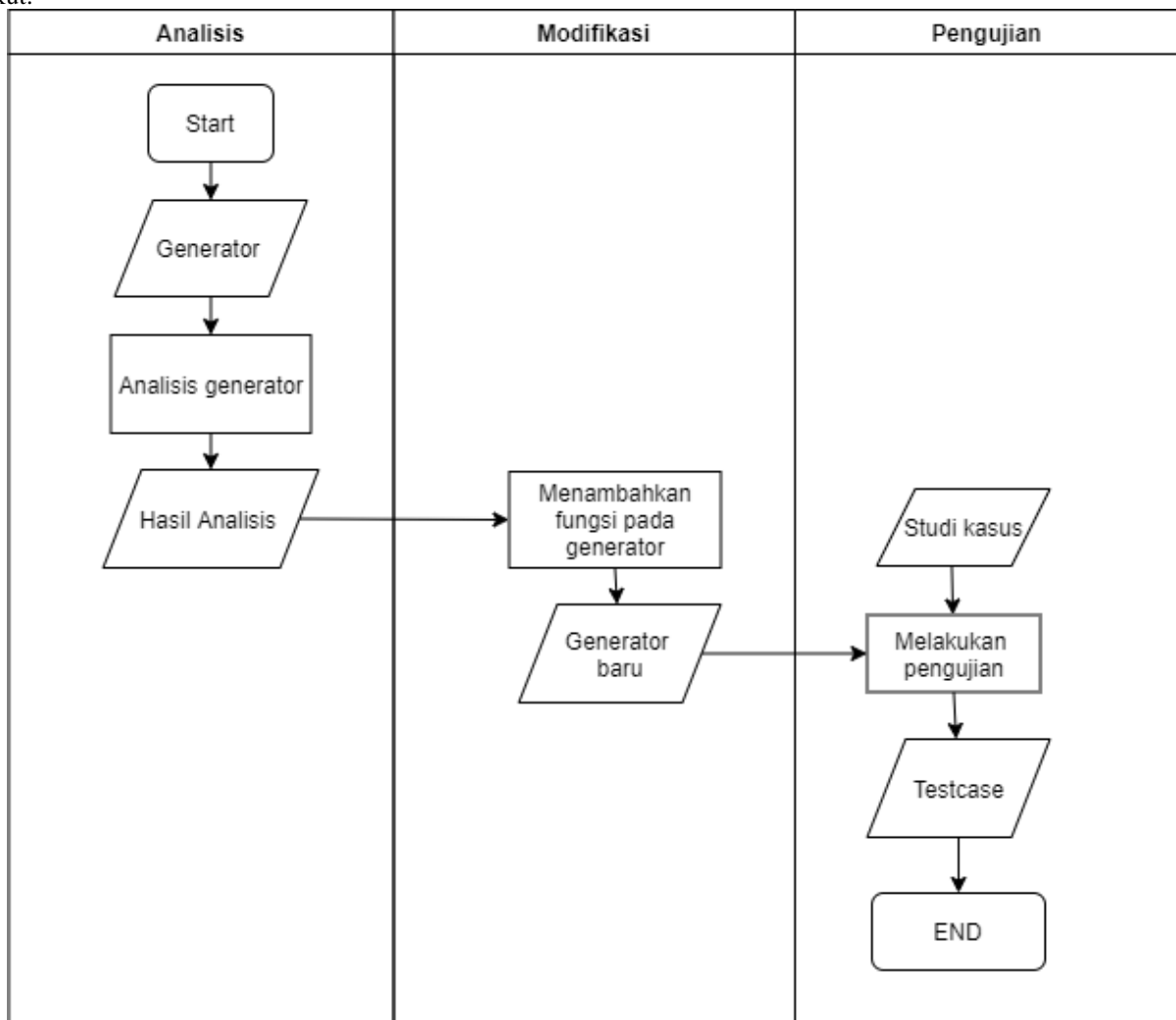
Dari tabel diatas, dapat dilihat bahwa penerapan Orthogonal array hanya berfokus pada 2 faktor input yaitu angka 1 dan 2. Istilah untuk rumusan ini adalah Orthogonal Array “strength 2”. Teknik ini sangat sesuai dengan pendekatan Pairwise testing yaitu melakukan semua kombinasi yang ekuivalen dengan 2 parameter input [9].

2.3. Generator Pairwise Testing

Pembangkit kasus uji dengan menerapkan suatu metode tertentu telah banyak tersedia secara online. Bahkan *source code* dalam bentuk *library* dan API dapat diakses dengan mudah. Termasuk dengan metode pairwise testing. Diantara *source code* yang untuk pairwise testing yang banyak digunakan walaupun tidak memiliki GUI adalah PICT (*Pairwise Independent Combinatorial Testing*), yaitu tool yang disediakan dari Microsoft sebagai pembangkit kasus uji dengan metode Pairwise. Namun, PICT hanya dapat dijalankan melalui command Prompt. Untuk melakukan tes menggunakan PICT, terdapat beberapa pengaturan yang perlu diterapkan terlebih dahulu, kemudian menyiapkan data tes dengan format yang telah ditentukan. Pembangkit kasus uji lainnya memiliki bervariasi aturan juga, seperti AETG yang diterbitkan oleh Telcordia, adalah pembangkit kasus uji *web-based* dan merupakan komersial. Untuk dapat menggunakan fungsi-fungsi tertentu pada pembangkit, pengguna harus berlangganan dan melakukan pembayaran. Pada penelitian ini, pembangkit yang digunakan berbentuk *source code* berbahasa python.

3. Sistem yang Dibangun

Dalam penelitian ini dilakukan modifikasi pada pembangkit kasus uji dengan metode pairwise testing. Modifikasi tersebut adalah menambahkan fungsi filter pada pembangkit kasus uji yang sudah ada. Penelitian berjalan dalam 3 jenis tahapan. Tahap pertama adalah analisis, yaitu membuka pembangkit kasus uji yang sudah ada, kemudian melakukan analisis terhadap pembangkit kasus uji tersebut. Hasil analisis yang telah dilakukan akan menjadi dasar tahap berikutnya yaitu modifikasi. Tahapan ini dilakukan pada pembangkit kasus uji dan menghasilkan pembangkit yang baru. Kemudian, untuk mengetahui apakah fungsi yang ditambahkan sudah sesuai atau tidak, dilakukan tahap pengujian dengan studi kasus. Adapun alur pada penelitian ini mulai dari membuka pembangkit kasus uji hingga menghasilkan pembangkit baru yang telah dimodifikasi dengan fungsi filter dalam bentuk *flowchart* adalah sebagai berikut:



1. Generator

Membuka *source code* pembangkit kasus uji yang telah tersedia menggunakan notebook yang sesuai dengan bahasa pemrograman generator. Dalam hal ini menggunakan Jupyter Anaconda.
2. Analisis generator

Melakukan analisis terhadap pembangkit kasus uji sebelum diterapkan modifikasi. Adapun metode analisis yang dilakukan pada tahap ini adalah metode analisis kualitatif untuk mengetahui kekurangan dari pembangkit yang sudah ada. Pada pembangkit tersebut, kasus uji yang dihasilkan telah sesuai dengan prinsip pairwise testing. Yaitu, jumlah test case yang dihasilkan tidak utuh hasil kombinasi. Namun, pembangkit tersebut tidak memiliki fungsi filter sebagai fungsi yang dapat memberikan batasan ketika akan melakukan generate test case. Berikut beberapa generator pairwise testing dengan
3. Hasil Analisis

Analisis mengacu pada pembangkit kasus uji yang sudah ada, pembangkit tersebut berupa Library dengan metode Pairwise testing untuk tes kombinasi yang dapat mengurangi jumlah data set. Analisis pembangkit dilakukan dengan melakukan pengamatan pada program library yang dimaksud. Kemudian dilakukan perbandingan fungsi dengan pembangkit kasus uji dengan metode pairwise testing yang tersedia secara online. Beberapa kelemahan pada library tersebut adalah:

 - a. Tidak memiliki fungsi filter untuk parameter value, yaitu ketika pengguna akan memberikan batasan pada suatu parameter input untuk tidak berinteraksi dengan parameter lainnya.
 - b. Tidak dapat melakukan fungsi tested pairs, yaitu fungsi untuk tidak melakukan kombinasi pada parameter yang sudah melakukan interaksi.
 - c. Tidak dapat melakukan triplewise testing dengan 3 faktor inputan sebagai parameter kombinasi.

Adapun kelebihan yang dimiliki oleh library ini adalah fungsi n-wise untuk melakukan kombinasi.
4. Menambahkan fungsi pada generator

Modifikasi dilakukan dengan menambahkan *code* program pada *source code* yang tersedia. Yaitu *code* fungsi filter menggunakan metode pairwise testing. Kode tersebut mengikuti aturan algoritma dasar percabangan dan pengulangan. Pada program yang ditambahkan, diberikan *constraint* waktu selama 10 menit apabila panjang input kolom lebih dari 5. Hal ini perlu dilakukan karena *constraint* harus di spesifikasikan oleh penguji sebelum memulai pembangkitan. *Constraint* dapat di terapkan berupa aturan-aturan tertentu dan juga ekspresi logika. Ekspresi logika dan aturan menggambarkan sebuah kondisi yang harus terpenuhi setiap melakukan pengujian, dan berfungsi untuk validasi terhadap set data yang dihasilkan.
5. Generator baru

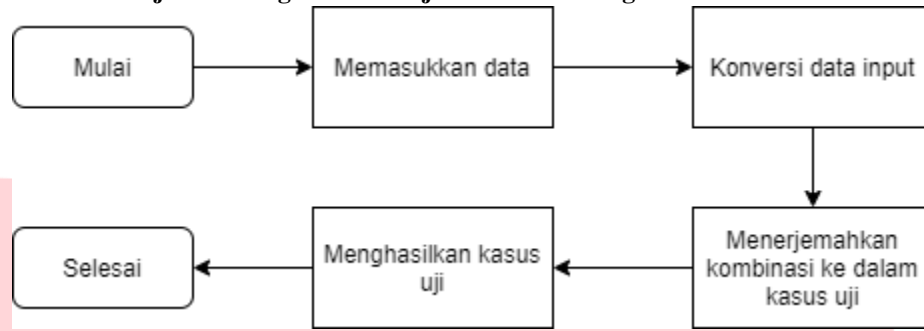
Pembangkit yang telah diberikan tambahan fungsi filter akan disebut dengan generator baru. Yaitu hasil dari modifikasi generator
6. Studi Kasus

Terdapat 2 jenis studi kasus yang digunakan pada pengujian pembangkit yaitu studi kasus dengan inputan 2x2x3 untuk melihat perbedaan fungsi ketika menggunakan studi kasus yang lebih besar. Yang kedua adalah studi kasus dengan inputan 4x4x4x5 yang berasal dari data Yayasan Abul Barokat NW Bunkate. Kedua jenis data digunakan sebagai pengujian fungsi filter karena keterbatasan terhadap penggunaan data dengan entri yang banyak. Kemudian, inputan data 2x2x3 harus menjadi data uji pada fungsi filter, dengan tujuan untuk dapat melihat dampak yang diberikan oleh fungsi pada data dengan kombinasi yang terbatas atau minimum.
7. Melakukan pengujian

Pengujian diperlukan untuk memastikan apakah fungsi filter telah berhasil ditambahkan pada generator atau tidak.
8. Testcase

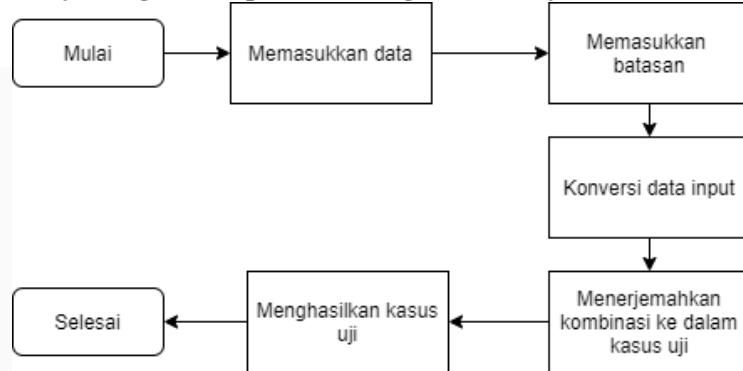
Set data yang dihasilkan oleh pengujian fungsi filter dengan studi kasus yang diberikan menjadi tolak ukur penerapan fungsi filter.

a. Alur Kerja Pembangkit Kasus Uji Pairwise Testing



1. Memasukkan data
Pembangkit kasus uji mulai bekerja dengan cara dimasukkan data yang akan dibentuk sebagai kasus uji.
2. Konversi data input
Kemudian data inputan akan di konversi secara numerik.
3. Menerjemahkan kombinasi ke dalam kasus uji
Setelah itu, dilakukan kombinasi dari inputan dan diterjemahkan menjadi kasus uji
4. Menghasilkan kasus uji
Pembangkit kasus uji menghasilkan kasus uji

b. Alur Kerja Fungsi Filter pada Pembangkit Kasus Uji Pairwise Testing



1. Memasukkan data
Pembangkit kasus uji mulai bekerja dengan cara dimasukkan data yang akan dibentuk sebagai kasus uji.
2. Memasukkan batasan
Dalam hal ini diberikan batasan antara inputan yang tidak dapat berinteraksi, sebagai acuan saat melakukan kombinasi
3. Konversi data input
Kemudian data inputan akan di konversi secara numerik.
4. Menerjemahkan kombinasi ke dalam kasus uji
Setelah itu, dilakukan kombinasi dari inputan dan diterjemahkan menjadi kasus uji sesuai dengan batasan yang berikan
5. Menghasilkan kasus uji
Pembangkit kasus uji menghasilkan kasus uji

6. Evaluasi

a. Hasil

Modifikasi dilakukan setelah melalui tahap analisis terlebih dahulu. Adapun kode modifikasi yang dilakukan terhadap pembangkit kasus uji mengikuti aturan penulisan algoritma sebagai berikut:

Kasus	Pseudocode
Sebuah data memiliki inputan: Versi: Version1, Version2, Version3, Version4 Jenis data: B, A Warna: white, black, yellow, blue Ukuran: 10, 20, 15, 30, 60 Selanjutnya akan dibangun set kasus uji berdasarkan inputan diatas dengan metode pairwise testing, namun diberikan filter jenis data "B" tidak bisa berjalan dengan "Version2" dan data "A" tidak bisa berjalan dengan "Version3"	<pre> Nama_fungsi: Deklarasi n if n > 1: if (jenis data B) dan (versi Version2) then return False if (jenis data A) dan (versi Version3) then return False return True end if </pre>

Kemudian dimasukkan data pada pembangkit kasus uji untuk mengetahui apakah modifikasi berjalan dengan baik atau tidak. Berikut data yang digunakan sebagai pengujian:

a. Data uji 2x2x3

Data uji diambil dari studi kasus pengujian pairwise testing atau data dengan minimum kombinasi. Data berjenis 2x2x3 umum digunakan untuk pengujian kombinasi agar dapat melihat pengurangan jumlah set data yang dihasilkan.

Nama Parameter	Parameter Input	Batasan
Type	Single, Span	Single tidak bisa bersama Slow
Format	Quick, Slow	Span tidak bisa bersama Quick
Size	100, 500, 1000	

Dengan menggunakan kombinasi yang utuh, maka kemungkinan data set yang didapatkan adalah sebanyak 12 set data:

Tabel 2 Kombinasi data 2x2x3

No	Type	Format	Size
1	Single	Quick	100
2	Single	Quick	500
3	Single	Quick	1000
4	Single	Slow	100
5	Single	Slow	500
6	Single	Slow	1000
7	Span	Quick	100
8	Span	Quick	500
9	Span	Quick	1000
10	Span	Slow	100
11	Span	Slow	500
12	Span	Slow	1000

Kemudian diterapkan pairwise testing pada kombinasi diatas, sehingga mengacu pada kombinasi 2 pair input saja, maka hasil yang di dapatkan adalah berupa 6 buat set data seperti pada data berikut:

PAIRWISE:

- 0: ['Single', 'Quick', 100]
- 1: ['Span', 'Slow', 100]
- 2: ['Span', 'Quick', 500]
- 3: ['Single', 'Slow', 500]
- 4: ['Single', 'Slow', 1000]
- 5: ['Span', 'Quick', 1000]

Semua kombinasi antara 2 faktor (Type dan Format) telah melakukan interaksi pada set data ke 0 (Single dan Quick) , 3 dan 4 (Single dan Slow), 2 dan 5 (Span dan Quick), 1 (Span dan Slow). Kemudian antara 2 faktor lainnya (Format dan Size) juga telah melakukan interaksi. Yaitu pada set data 0 (Quick dan 100), 2 (Quick dan 500), 5 (Quick dan 1000), 1 (Slow dan 100), 3 (Slow dan 500), 4 (Slow dan 1000). Begitu juga dengan interaksi antara faktor (Type dan Size), kemungkinan dari kombinasi yang terjadi sudah melakukan interaksi. Oleh karena itu, jumlah test case yang dihasilkan pada data 2x2x3 menggunakan pairwise testing adalah 6 buah test case. Mengacu pada tabel 2, set data yang dihasilkan dari kombinasi utuh pada pairwise testing adalah kombinasi nomor 1,5,6,8,9 dan 10.

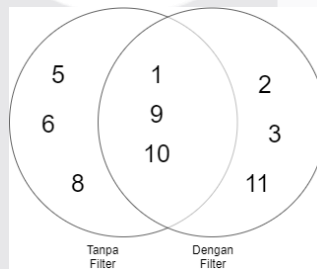
Adapun set data yang tidak dihasilkan atau di eliminasi oleh pairwise testing pada *tabel 2* adalah set data nomor 2,3,4,7,11, dan 12. Pengurangan jumlah set data menjadi setengah jumlah kombinasi yang seharusnya, menjadi alasan mengapa Pairwise testing dapat digunakan untuk membuat set data tanpa mengurangi *coverage* dari input parameter yang diinginkan.

Kemudian, data yang sama dibangkitkan dengan pembangkit yang telah ditambahkan fungsi filter, dengan batasan Single tidak bisa bersama Slow dan Span tidak bisa bersama Quick. Maka set yang dihasilkan adalah sebagai berikut:

PAIRWISE:

- 0: ['Single', 'Quick', 100]
- 1: ['Span', 'Slow', 100]
- 2: ['Span', 'Slow', 500]
- 3: ['Single', 'Quick', 500]
- 4: ['Single', 'Quick', 1000]
- 5: ['Span', 'Slow', 1000]

Jumlah testcase yang dihasilkan dengan fungsi filter dan tanpa fungsi filter adalah sama. Berdasarkan validasi tanpa menggunakan filter, set data dengan pairwise testing memberikan hasil 6 buah testcase, yaitu nomor 1,5,6,8,9 dan 10 (mengacu pada tabel 2). Namun dengan menggunakan filter, data yang dihasilkan adalah nomor 1,2,3,9,10 dan 11. Dalam diagram dapat dilihat sebagai berikut:



Data yang sama hanya 3 buah test case, kemudian 3 lainnya mengikuti batasan filter yang diberikan. Pada hasil testcase tanpa filter, data yang tidak masuk dalam hasil dengan filter adalah data nomor 5 (Single, Slow, 500), 6 (Single Slow 1000) dan 8 (Span, Quick, 500). Hal ini karena fungsi filter memberikan batasan bahwa Single tidak bisa bersama Slow, oleh karena itu data nomor 5 dan 6 tidak ada. Kemudian batasan berikutnya adalah Span tidak bisa bersama Quick, maka data nomor 10 tidak ada. Namun, agar tetap mengikuti aturan Pairwise testing dan Orthogonal Array bahwa setiap kombinasi harus melakukan interaksi pada 2 faktor.

Maka, untuk melengkapi interaksi antara 2 faktor (Type dan Format), (Format dan Size), dan (Type dan Size) hasil yang diperoleh diisi dengan sisa dari luar batasan. Oleh karena itu, jumlah test case yang dihasilkan dengan filter dan tanpa filter adalah sama.

b. Data uji 4x4x4x5

Data uji diambil dari data Yayasan Abul Barokat Wannafahat NW Bunkate

Nama Parameter	Parameter Input	Batasan
Jenjang	SMP, SMA, MA, SMK	SMK tidak bisa mengambil Hafalan 1
Jurusan	Hafalan1, Hafalan2, Tilawah, Bahasa	SMP tidak bisa mengambil Tilawah
Status Masuk	Baru, Pindahan, Lanjutan, Alumni	
Asal Daerah	LombokTimur, LombokBarat, LombokTengah, LombokUtara, LombokSelatan	

Hasil yang didapatkan adalah:

PAIRWISE:

```

0: ['SMP', 'Hafalan1', 'Baru', 'Lombok Timur']
1: ['SMA', 'Hafalan2', 'Pindahan', 'Lombok Timur']
2: ['MA', 'Tilawah', 'Lanjutan', 'Lombok Timur']
3: ['SMK', 'Bahasa', 'Alumni', 'Lombok Timur']
4: ['SMK', 'Tilawah', 'Pindahan', 'Lombok Barat']
5: ['MA', 'Hafalan2', 'Baru', 'Lombok Barat']
6: ['SMA', 'Hafalan1', 'Alumni', 'Lombok Barat']
7: ['SMP', 'Bahasa', 'Lanjutan', 'Lombok Barat']
8: ['SMP', 'Hafalan2', 'Alumni', 'Lombok Tengah']
9: ['SMA', 'Tilawah', 'Baru', 'Lombok Tengah']
10: ['MA', 'Bahasa', 'Pindahan', 'Lombok Tengah']
11: ['SMK', 'Hafalan2', 'Lanjutan', 'Lombok Utara']
12: ['SMK', 'Bahasa', 'Baru', 'Lombok Selatan']
13: ['MA', 'Hafalan1', 'Lanjutan', 'Lombok Selatan']
14: ['SMA', 'Bahasa', 'Lanjutan', 'Lombok Utara']
15: ['SMP', 'Hafalan1', 'Pindahan', 'Lombok Utara']
16: ['SMP', 'Hafalan2', 'Alumni', 'Lombok Selatan']
17: ['MA', 'Tilawah', 'Alumni', 'Lombok Utara']
18: ['SMK', 'Tilawah', 'Lanjutan', 'Lombok Tengah']
19: ['SMK', 'Tilawah', 'Pindahan', 'Lombok Selatan']
20: ['SMK', 'Tilawah', 'Baru', 'Lombok Utara']

```

Hasil yang didapatkan tanpa menggunakan fungsi filtering adalah sebagai berikut:

PAIRWISE:

```

0: ['SMP', 'Hafalan1', 'Baru', 'Lombok Timur']
1: ['SMA', 'Hafalan2', 'Pindahan', 'Lombok Timur']
2: ['MA', 'Tilawah', 'Lanjutan', 'Lombok Timur']
3: ['SMK', 'Bahasa', 'Alumni', 'Lombok Timur']
4: ['SMK', 'Tilawah', 'Pindahan', 'Lombok Barat']
5: ['MA', 'Hafalan2', 'Baru', 'Lombok Barat']
6: ['SMA', 'Hafalan1', 'Alumni', 'Lombok Barat']
7: ['SMP', 'Bahasa', 'Lanjutan', 'Lombok Barat']
8: ['SMP', 'Tilawah', 'Alumni', 'Lombok Tengah']
9: ['SMA', 'Bahasa', 'Baru', 'Lombok Tengah']
10: ['MA', 'Hafalan1', 'Pindahan', 'Lombok Tengah']
11: ['SMK', 'Hafalan2', 'Lanjutan', 'Lombok Tengah']
12: ['SMK', 'Hafalan1', 'Lanjutan', 'Lombok Utara']

```

```
13: ['MA', 'Bahasa', 'Alumni', 'Lombok Utara']
14: ['SMA', 'Tilawah', 'Baru', 'Lombok Utara']
15: ['SMP', 'Hafalan2', 'Pindahan', 'Lombok Utara']
16: ['SMA', 'Hafalan2', 'Alumni', 'Lombok Selatan']
17: ['SMK', 'Bahasa', 'Pindahan', 'Lombok Selatan']
18: ['SMK', 'Tilawah', 'Baru', 'Lombok Selatan']
19: ['SMA', 'Hafalan1', 'Lanjutan', 'Lombok Selatan']
20: ['SMP', 'Hafalan1', 'Lanjutan', 'Lombok Selatan']
21: ['MA', 'Hafalan1', 'Lanjutan', 'Lombok Selatan']
```

Pada data yang sedikit lebih banyak, terjadi perbedaan jumlah testcase yang dihasilkan. Apabila menggunakan fungsi filter, testcase yang dihasilkan adalah 19 set data. Namun, apabila tanpa menggunakan fungsi filter, set data yang dihasilkan berjumlah 21. Hal ini menunjukkan bahwa pairwise testing akan semakin memberikan dampak yang kuat apabila digunakan pada data input yang banyak. Begitu pula dengan penggunaan fungsi filter, semakin banyak batasan yang diberikan, semakin sedikit hasil test case yang dihasilkan oleh pembangkit.

b. Analisis Hasil Pengujian

RQ1. Apakah fungsi filter yang ditambahkan pada generator kasus uji sudah mampu melakukan filtering terhadap testcase?

Pengujian dilakukan dengan menggunakan 2 jenis data dengan batasan masing-masing. Berdasarkan hasil pengujian dengan batasan yang diberikan, pembangkit berhasil membuat kasus uji sesuai dengan mengikuti batasan. Hal ini menunjukkan fungsi filter sudah mampu melakukan filtering pada data. Pada pengujian data pertama diberikan batasan inputan “Single” tidak bisa bersama “Slow” dan “Span” tidak bisa bersama “Quick”. Kemudian pada bagian hasil pembangkit menghasilkan 5 buah kasus uji untuk parameter input diantaranya tidak ada kasus uji “Single” dengan “Slow” dan tidak ada juga “Span” dengan “Quick”. Begitupun pada pengujian kedua. Batasan yang diberikan adalah “SMK” tidak bisa mengambil jurusan “Hafalan1” dan “SMP” tidak bisa berasal dari status “Lanjutan”. Data input tersebut menghasilkan 19 set kasus uji dengan mengikuti batasan yang diberikan.

RQ2. Bagaimana perbedaan hasil testcase dengan filtering dan tanpa filtering?

Hasil kasus uji dengan menerapkan fungsi filter berhasil mengikuti batasan yang diberikan. Jumlah kasus uji yang dihasilkan telah mengikuti aturan pairwise testing yang menerapkan Orthogonal Array sehingga pembangkit menghasilkan kasus uji yang sedikit. Pada studi kasus yang pertama, dengan data 2x2x3 jumlah testcase yang dihasilkan oleh pembangkit dengan filter dan tanpa filter berjumlah sama yaitu 5 buah set data. Hal ini membuktikan bahwa pairwise testing kurang memiliki dampak pada data dengan input yang kecil atau sedikit. Kemudian, pada studi kasus dengan data 4x4x4x5 jumlah testcase yang dihasilkan sedikit memiliki perbedaan antara pembangkit menggunakan filter dan tanpa filter. Hal ini menunjukkan bahwa data setelah menerapkan filtering berkurang sesuai dengan aturan atau batasan yang diberikan pada parameter.

RQ3. Bagaimana proses eliminasi set data yang dihasilkan oleh fungsi filter?

Proses eliminasi yang dilakukan oleh fungsi filter, mengacu kepada batasan atau filter yang diberikan pada program tanpa menghilangkan teknik pairwise testing dengan melakukan interaksi antara 2 faktor. Sehingga pada kasus data uji 2x2x3, jumlah test case dengan menggunakan filter dan tanpa filter menghasilkan jumlah yang sama.

7. Kesimpulan

Berdasarkan hasil pengujian yang dilakukan. Modifikasi terhadap pembangkit kasus uji dengan metode pairwise testing berhasil diterapkan. Sehingga kasus uji yang dihasilkan oleh pembangkit mengikuti batasan yang diberikan. Namun, dalam hal ini kasus uji yang digunakan relatif kecil. Karna pairwise testing akan lebih memberikan dampak apabila digunakan pada data yang besar dan banyak.

Referensi

- [1] F. K. Mohd. Ehmer Khan, "A Comparative Study of White Box, Black Box and Grey Box Testing Techniques," *(IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 3, no. 6, p. 1, 2012.
- [2] Y. Badzhieva, "What Is Pairwise Testing And How To Use It For Cost-Effective QA," 22 April 2019.
- [3] O. O. Y. J. S. Salamah, "Using pairwise testing to verify automatically generated formal specifications," *2015 IEEE 16th International Symposium on High Assurance Systems Engineering*, pp. 279-280, 2015.
- [4] L. E. a. E. E. Peter Charbachi, "Can Pairwise Testing Perform Comparably to Manually Handcrafted Testing Carried Out by Industrial Engineers?," *2012017 IEEE International Conference on Software Quality, Reliability and Security (Companion Volume)*, p. 1, 2017.
- [5] T. S. Sompong Nakornburi, "A Tool for Constrained Pairwise Test Case Generation Using Statistical User Profile Based Prioritization," *13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, p. 1, 2016.
- [6] S. W. J. H. H. Pungki Nurhudha, "Pembangunan Kakas Untuk Menghasilkan Kasus Uji Dengan Metode Pairwise Testing," *Sarjana Informatika Telkom University*, 2020.
- [7] A. A. A. a. K. Z. Z. Abdullah B. Nasser, "Adopting Search-Based Algorithms for Pairwise Testing," *2015 4th International Conference on Software Engineering and Computer Systems (ICSECS)*, Vol. %1 dari %2 August 19-21, p. 1, 2015.
- [8] N. M. Ljubomir Lazic, "Orthogonal Array application for optimal combination of software defect detection techniques choices," *WSEAS TRANSACTIONS on COMPUTERS*, 2014.
- [9] labs.sogeti.com, "Orthogonal arrays and pairwise testing," <https://www.tmap.net/wiki/orthogonal-arrays-and-pairwise-testing>, Akses 15 September 2021.