

Pembangunan Kakas Untuk Menghasilkan Kasus Uji Dengan Metode *Pairwise Testing*

Pungki Nurhudha¹, Sri Widowati², Jati H. Husen³

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

¹pungkinurhudha@students.telkomuniversity.ac.id, ²sriwidowati@telkomuniversity.ac.id,

³jatihusen@telkomuniversity.ac.id

Abstrak

Dalam pengujian perangkat lunak, salah satu yang harus dimiliki yaitu kasus uji. kasus uji berperan penting dalam pengujian perangkat lunak. Kasus uji dapat dihasilkan dengan menerapkan metode *Pairwise Testing*. *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. Pada penerapannya, metode *Pairwise Testing* cukup kompleks karena memiliki tahapan yang panjang untuk dapat menghasilkan kasus uji. Maka diperlukan kakas untuk menghasilkan kasus uji menggunakan metode *Pairwise Testing*. Kakas yang telah dibangun maka perlu dilakukan uji coba, untuk membuktikan apakah kakas bisa menghasilkan kasus uji yang sesuai dengan metode *Pairwise Testing*. Setelah dilakukan uji coba kakas maka ditemukan bahwa tidak semua hasil dari kakas sesuai, dan tidak semua pengujian dapat menerapkan metode *Pairwise Testing* dengan baik.

Kata kunci: kasus uji, *pairwise testing*

Abstract

In software testing, one that must have is a test case. Test cases play an important role in software testing. Test cases can be generated by applying the *Pairwise Testing* method. *Pairwise Testing* is a combinatorial testing technique that seeks to ensure that test cases focus on defining a test set that provides a variable pair test, not a test that combines all variables. In practice, the *Pairwise Testing* method is quite complex because it has a long stage to produce test cases. So tools are needed to produce test cases using the *Pairwise Testing* method. Tools that have been built need to be tested, to prove whether the tools can produce test cases that are in accordance with the *Pairwise Testing* method. After testing the tools, it was found that not all results from the tools were appropriate, and not all testers were able to apply the *Pairwise Testing* method properly.

Keywords: test case, *pairwise testing*

1. Pendahuluan

Latar Belakang

Dalam pengujian perangkat lunak, dibutuhkan kasus uji untuk menggambarkan jenis subjek yang diuji dan objek yang akan diuji [1]. Aspek penting dari pengujian adalah proses yang digunakan untuk merencanakan dan mengimplementasikannya [2]. Metode yang dapat menghasilkan kasus uji adalah metode *Pairwise Testing*. *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]. Metode tersebut cukup kompleks penerapannya, sehingga metode *Pairwise Testing* jarang digunakan untuk menghasilkan kasus uji [4][5].

Pairwise Testing didasarkan pada pengamatan bahwa kesalahan perangkat lunak sering melibatkan interaksi antar parameter. Untuk menerapkan metode *Pairwise Testing* perlu untuk menemukan *input* dan dapat mencocokkan setiap set parameter. Keduanya adalah masalah yang sulit, tetapi kedua masalah tersebut dapat diselesaikan dengan algoritma yang dapat disebut kakas *Pairwise Testing*[6]. Maka dengan membangun kakas *Pairwise Testing* dapat meningkatkan efektivitas dari penggunaan waktu dan membantu proses pembuatan kasus uji. Pembangunan kakas *Pairwise Testing* untuk menghasilkan kasus uji dapat menggunakan *Orthogonal Array* agar kasus uji yang dihasilkan sesuai. Proses *Orthogonal Array* yang menentukan penyusunan kasus uji. *Orthogonal Array* dapat dilihat sebagai rencana percobaan *multifactor* dimana kolom sesuai dengan faktor, entri dalam kolom sesuai dengan tingkat uji faktor dan baris sesuai kasus uji [7]. Proses yang dilakukan *Orthogonal Array* menentukan bagian kasus uji yang akan ditampilkan dengan jumlah yang sedikit tetapi mempunyai hasil yang akurat.

Metode *Pairwise Testing* yang digunakan pada penelitian ini hanya sampai menghasilkan kasus uji untuk selanjutnya bisa dilakukan pengujian pada perangkat lunak yang dijadikan kasus uji. Pada penelitian ini juga terdapat kontribusi yaitu, mengusulkan kakas *Pairwise Testing* untuk pengembangan hasil kasus uji yang lebih baik. Lalu, peneliti dapat memahami kelebihan dan kekurangan metode *Pairwise Testing* untuk menghasilkan kasus uji.

Rumusan Masalah

Berdasarkan penjelasan kondisi dan beberapa masalah pada latar belakang, penelitian ini memiliki pertanyaan penelitian atau *Research Question* (RQ) untuk memecahkan masalah dalam proses menghasilkan kasus uji dengan metode *Pairwise Testing* yaitu:

1. Apakah kasus *Pairwise Testing* yang dibangun sudah tepat?
2. Bagaimana hasil kasus uji dengan menggunakan kasus *Pairwise Testing* dari hasil pengujian?

Tujuan

Berdasarkan rumusan masalah, tujuan dari penelitian ini adalah membuat dan menganalisis kasus untuk dapat menghasilkan kasus uji dengan metode *Pairwise Testing*.

Batasan Masalah

Ada beberapa Batasan masalah pada penelitian ini pertama, diterapkan untuk menghasilkan kasus uji mempunyai keterbatasan orang untuk melakukan tes pada kasus. Kedua, tidak bisa mengawasi langsung orang yang menggunakan kasus ini. Lalu, metode *pairwise testing* hanya bisa digunakan pada studi kasus yang banyak entri dan konfigurasi pengujian yang besar. Misalnya, jika sedang mengerjakan situs web / aplikasi web yang dapat diunduh & digunakan oleh konsumen dari seluruh dunia, agar perlu memastikan bahwa itu berfungsi dengan lancar di berbagai browser / perangkat / sistem operasi.

Organisasi Tulisan

Pada bagian 2 akan menjelaskan studi literatur yang sudah dilakukan, studi literatur berisi teori yang dapat dijadikan acuan mengenai *Pairwise Testing* dan teori mengenai kasus uji coba yang digunakan. bagian 3 menjelaskan sistem yang dibangun beserta skemanya, pada skema atau proses yang dibuat menjelaskan bagaimana menghasilkan kasus uji dengan metode *Pairwise Testing*.

2. Studi Terkait

2.1 Pairwise

Salah satu metode pengujian perangkat lunak yaitu *pairwise testing*. *pairwise testing* telah menjadi metode yang sangat diperlukan dalam alat pengujian perangkat lunak, teknik ini telah dikenal selama hampir 20 tahun [4]. *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]. Metode *Pairwise Testing* dapat meminimalkan jumlah tes yang dihasilkan agar lebih efisien waktu proses pengujian. [4]. *Pairwise Testing* adalah teknik yang bertujuan secara signifikan, mengurangi jumlah kasus uji yang diperlukan untuk menguji sistem perangkat lunak tertentu sambil memberikan jaminan cakupan yang memadai dari ruang masalah [3]. Set tes yang menyediakan tes pasangan variabel ini bermaksud bahwa metode *Pairwise Testing* tidak mengeluarkan semua pasangan yang dipasangkan melainkan hanya pasangan kasus uji yang diperlukan saja untuk menguji sistem perangkat lunak.

2.2 Kasus Uji

Untuk dapat melakukan proses pengujian, uji kasus dirancang, diimplementasikan dan digunakan untuk menguji sebagian atau seluruh aplikasi(subjek yang diuji). Hasil dari kasus uji ini kemudian dianalisis dan dievaluasi. Sebuah uji kasus digunakan disini untuk menggambarkan jenis subjek yang diuji dan item yang akan diuji[1]. Desain kasus uji tidak diamati atau dibatasi, tetapi upaya yang digunakan dicatat [8]. Menghasilkan kasus uji yang dimaksud pada metode *Pairwise Testing* hanya sampai mendapatkan kasus uji saja, tidak sampai mendapatkan hasil dari kasus uji tersebut. Untuk mengetahui hasil yang diharapkan dari kasus uji metode *pairwise testing* adalah setiap nilai dari terpasang dengan nilai dari parameter lainnya.

2.3 Orthogonal Array

Orthogonal Array merupakan bagian proses *Pairwise Testing* dengan kasus. Orthogonal Array dapat dilihat sebagai rencana percobaan *multifactor* dimana kolom sesuai dengan faktor, entri dalam kolom sesuai dengan tingkat uji factor dan baris sesuai uji kasus[7]. Sebagai contoh tabel Orthogonal Array adalah sebagai berikut

Row No.	Column No.		
	1	2	3
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0

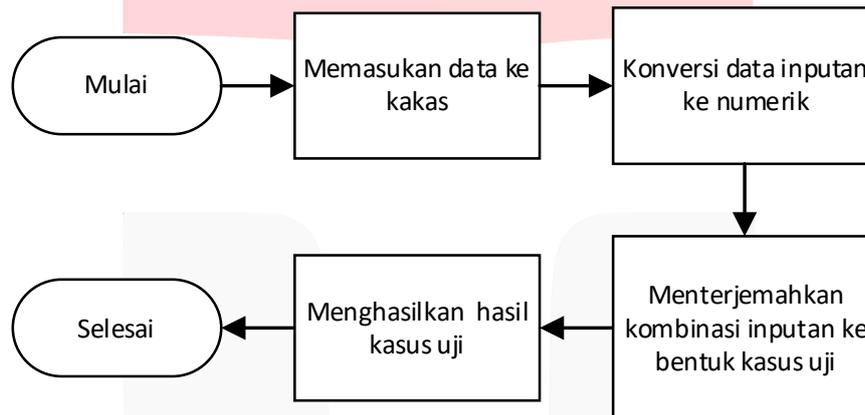
Gambar 1. Orthogonal Array OA₄ (2³) [7]

Orthogonal Array yang telah ditemukan dapat menghasilkan jumlah minimum skenario pengujian dan memberikan cakupan maksimum ruang variabel uji kasus[7].

3. Sistem yang Dibangun

Dalam penelitian ini dibangun kakas untuk membuktikan bahwa kakas yang telah dibangun menggunakan metode *Pairwise Testing* dapat digunakan sesuai fungsinya. Adapun langkah-langkah atau diagram balok dari kakas *Pairwise Testing* dalam penelitian ini. Pertama dari memasukan data ke kakas, hingga mendapatkan hasil kasus uji yang nantinya dapat disimpulkan berdasarkan waktu dan hasil evaluasi metode *pairwise testing*. Berikut ini konsep pengujian dengan kakas dari penelitian ini.

3.1 Konsep Kakas *Pairwise Testing* Untuk Menghasilkan Kasus Uji



Gambar 2. Proses Kakas *Pairwise Testing* Menghasilkan Kasus Uji

Kakas yang dibangun untuk melakukan pengujian *pairwise testing* bertujuan untuk mendukung evaluasi kebermanfaatan metode *pairwise testing*. Penjelasan proses dari skema pengujian adalah sebagai berikut:

- Masukan data ke kakas
Proses awal yaitu memasukan data sesuai dengan ukuran kombinasi yang diinginkan ke kakas *Pairwise Testing* dengan cara mengetik pada kolom yang sudah disediakan pada kakas.
- Konversi data inputan ke numerik
Pada proses ini pengubahan data kombinasi studi kasus akan diubah ke angka. Angka yang dimasukan mulai dari angka 0 hingga sebanyak value dari kombinasi yang dimasukan setiap kolomnya, seperti yang ada pada gambar 3.

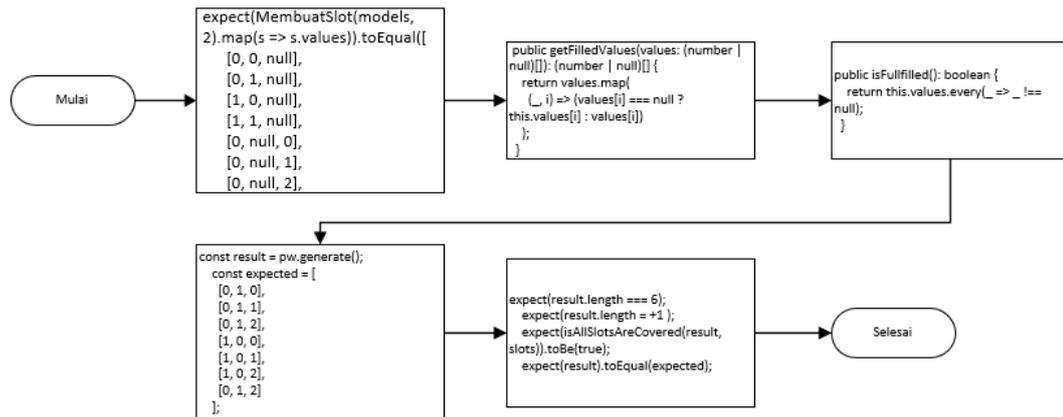
		AB	AC	BC
		00	00	00
A: 0, 1		01	01	01
B: 0, 1	translates to	10	02	02
C: 0, 1, 2		11	10	10
		11	11	11
		12	12	12

Gambar 3. Ilustrasi Cara Kerja Kakas *Pairwise Testing* [7]

- Menerjemahkan kombinasi inputan ke bentuk kasus uji
Proses ini melakukan penempatan data kombinasi awal ke bentuk kasus uji sesuai dengan atribut dari masing-masing kolom(gambar 3).
- Menghasilkan hasil kasus uji
Proses ini yaitu data yang dihasilkan dari kaskas *pairwise testing* dapat terlihat oleh penguji.

3.2 Cara Kerja Orthogonal Array

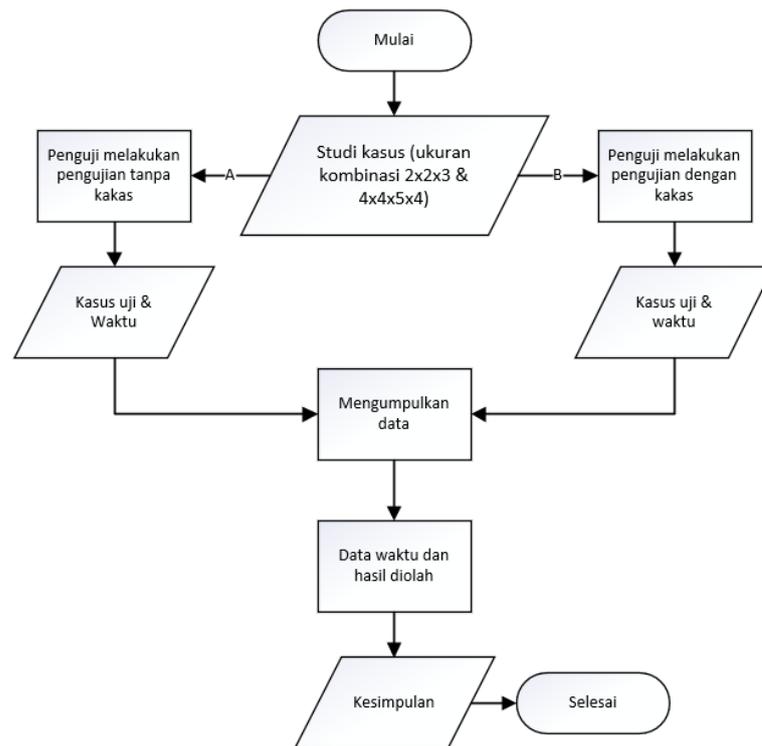
Tahapan cara kerja Orthogonal Array yaitu sebagai berikut



Gambar 4. Cara kerja Orthogonal Array Pada Kakas *Pairwise Testing*

- **Membuat Slot**
Sintaks bagian ini adalah tahapan pertama Orthogonal Array, sintaks ini berfungsi membuat tempat untuk setiap value dari parameter yang telah dimasukan ke kakas
- **Get filled values**
Pada bagian ini, sintaks berfungsi sebagai pengisi dari sintaks ‘membuat slot’
- **Isi full filled**
Sintaks isi full filled ini untuk memastikan bahwa slot telah terisi benar
- **Generate**
Sintaks generate ini adalah untuk membuat studi kasus yang dimasukan ditempatkan sesuai aturan Orthogonal Array.
- **Menggubah dan menampilkan**
Setelah ditempatkan setiap value nya, maka value tersebut ditampilkan pada halaman kakas setelah ditekan tombol ‘generate’.

3.3 Skema Pengujian Kakas *Pairwise Testing*



Gambar 5. Skema pengujian kakas *Pairwise Testing*

Pada bagian skema pengujian kakas *Pairwise Testing* menjelaskan bagaimana cara pengujian untuk mendapatkan hasil kasus uji

- **Studi kasus**
Terdapat 2 ukuran studi kasus untuk mencoba kakas *Pairwise Testing* ini. Studi kasus yang digunakan pada penelitian ini menggunakan sebagian dialog pada menu Microsoft Word sebagai contoh untuk kasus uji.
- **Pengujian dilakukan tanpa kakas**
Pada proses ini, pengujian melakukan *generate* kasus uji tanpa menggunakan bantuan kakas terlebih dahulu. Proses ini membuat pengujian kesulitan untuk mendapat kasus uji, karena proses tanpa kakas ini membutuhkan proses yang panjang.
- **Pengujian dilakukan dengan kakas**
Proses pengujian melakukan *generate* kasus uji dengan kakas ini menggunakan data studi kasus yang sama pada saat tanpa menggunakan kakas.
- **Kasus uji dan waktu**
Data kasus uji akan disimpan dan waktu saat pengerjaan disimpan untuk dikumpulkan dan bisa disimpulkan.
- **Mengumpulkan data**
Pada proses mengumpulkan data, data yang dikumpulkan adalah hasil kasus uji dari yang menggunakan kakas maupun yang tidak menggunakan kakas dan waktu saat melakukan *generate* kasus uji.
- **Data waktu dan hasil diolah**
Pengolahan data waktu dan hasil dengan cara ditinjau penggunaan waktu saat menggunakan kakas dan tidak menggunakan kakas. Lalu untuk pengolahan data hasil ditinjau dengan cara menyamakan dengan *Orthogonal Array* yang ukurannya sama dengan studi kasusnya.
- **Kesimpulan**
Setelah seluruh proses dilakukan, maka akan didapatkan kesimpulan saat menggunakan metode *Pairwise Testing*.

3.4 Data Pengalaman Penguji

Untuk melakukan *generate* kasus uji dengan *Pairwise Testing* membutuhkan beberapa penguji agar dapat menghasilkan data hasil uji. Pada penelitian ini dibantu oleh 4 orang mahasiswa Informatika yang telah mempunyai pengalaman mengenai pengujian dan memahami mengenai kasus uji pada saat kuliah atau saat melakukan kerja praktek, berikut tabel pengalaman penguji pada penelitian ini :

Tabel 1. Daftar Pengalaman Penguji

Penguji	Pengalaman	Status penguji
Penguji 1	Pernah melakukan Pengujian Secara otomatis, dan proses pengujian secara umum di perusahaan <i>game</i> (Agate)	Mahasiswa Universitas Telkom, tingkat 4
Penguji 2	Pernah melakukan pengujian <i>website</i> dengan metode menggunakan selenium	Mahasiswa Universitas Telkom, tingkat 4
Penguji 3	Pernah melakukan <i>Usability Testing</i> di salah satu produk aplikasi jual beli online	Mahasiswa Universitas Telkom, tingkat 4
Penguji 4	Pernah melakukan pengujian <i>website</i> dengan menggunakan selenium, pernah melakukan pengujian <i>website</i> perusahaan asuransi kesehatan dengan usability testing	Mahasiswa Universitas Telkom, tingkat 4

4. Evaluasi

4.1. Hasil

Penelitian dengan metode *Pairwise Testing* ini dibantu oleh 4 orang penguji yang dapat dilihat pada tabel 1. Penguji dipilih dengan syarat sudah pernah melakukan pengujian perangkat lunak di mata kuliah maupun pengalaman pengujian perangkat lunak lainnya. Penelitian ini menggunakan 2 ukuran yang berbeda agar mendapatkan data yang cukup untuk melakukan evaluasi. Berikut hasil dari penelitian ini:

A. Pengujian dengan ukuran 2x2x3

Pada pengujian ini menggunakan ukuran 2x2x3 bertujuan agar penguji dapat mempunyai pengalaman langsung dengan metode *pairwise testing*, karena jika langsung menggunakan ukuran yang besar akan membuat penguji kurang memahami. Berikut hasil penghitungan waktu pengujian untuk 2x2x3

Tabel 2. Waktu Menghasilkan Kasus Uji Tanpa Kakas

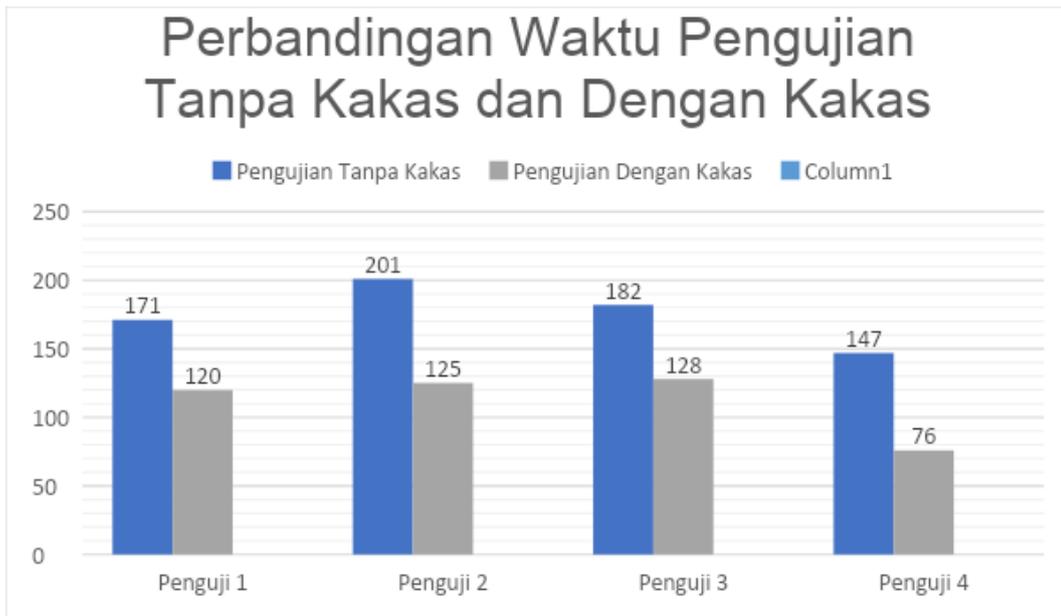
No	Penguji	Waktu Pengujian
1	Penguji 1	171 detik
2	Penguji 2	201 detik
3	Penguji 3	182 detik
4	Penguji 4	147 detik

Perhitungan waktu pengujian tanpa kakas terlihat bahwa setiap penguji memiliki waktu yang berbeda-beda dalam melakukan pengujian. Dapat disimpulkan bahwa pengujian tanpa kakas pada tabel 2 menggunakan waktu lebih dari 140 detik.

Tabel 3. Waktu Menghasilkan Kasus Uji Dengan Kakas

No	Penguji	Waktu Pengujian
1	Penguji 1	120 detik
2	Penguji 2	125 detik
3	Penguji 3	128 detik
4	Penguji 4	53 detik

Perhitungan waktu pengujian dengan kakas juga memiliki perbedaan setiap penguji, akan tetapi pengujian dengan kakas ini lebih cepat dibanding tanpa kakas, seluruh penguji dapat menempuh pengujian hingga selesai dengan waktu dibawah 140 detik. Berikut adalah grafik perbedaan durasi waktu setiap penguji.



Gambar 6. Perbandingan waktu Menghasilkan Kasus Uji tanpa kakas dan dengan kakas

Dari hasil perbandingan waktu pengujian *pairwise testing* dengan kakas dan tanpa kakas pada ukuran 2x2x3 dapat dilihat pada gambar 4. Jadi dapat dihitung persentase perbandingan setiap penguji dengan menghitung waktu pengujian tanpa kakas dibagi waktu pengujian dengan kakas lalu dikali 100 setiap pengujinya. Berikut hasil perhitungan persentasenya

Tabel 4. Persentase Saat Menghasilkan Kasus Uji Tanpa Kakas dan Dengan Kakas Ukuran 2x2x3

No	Penguji	Persentase perbandingan
1	Penguji 1	142,5%
2	Penguji 2	160%
3	Penguji 3	142,1%
4	Penguji 4	193,4%

B. Pengujian dengan ukuran 4x4x4x5

Pengujian kedua ini menggunakan ukuran yang ukurannya lebih besar karena dianggap penguji sudah lebih memahami metode *pairwise testing*. Pengujian kedua ini hanya berbeda pada ukuran *test case* dari menu font Microsoft Word. Berikut hasil pencatatan data waktu:

Tabel 5. Waktu Menghasilkan Kasus Uji Tanpa Kakas

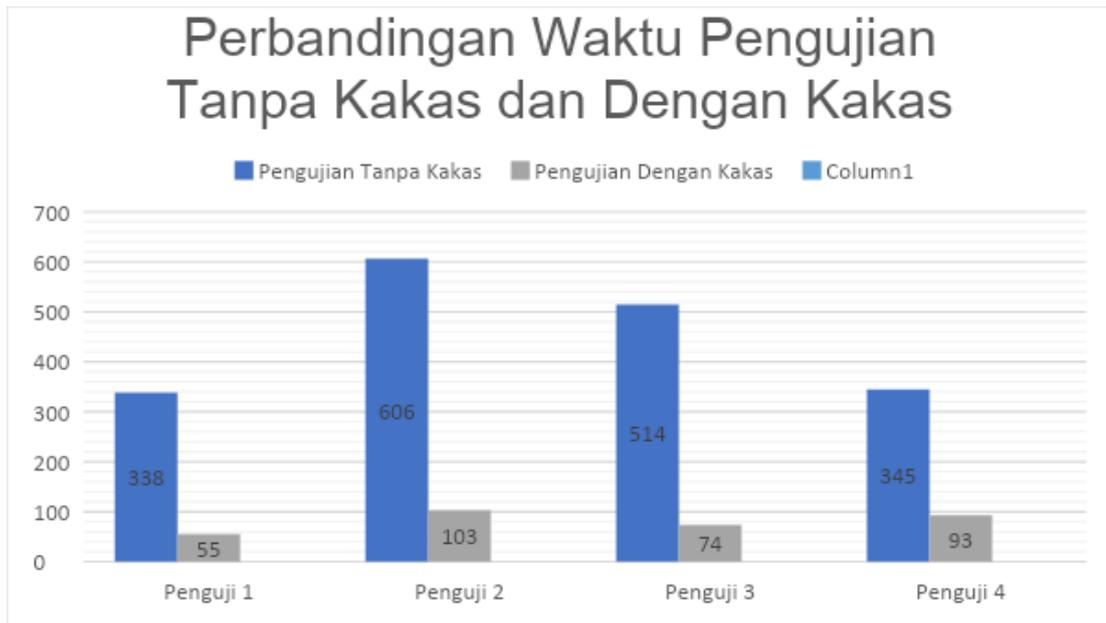
No	Penguji	Waktu Pengujian
1	Penguji 1	338 detik
2	Penguji 2	606 detik
3	Penguji 3	514 detik
4	Penguji 4	345 detik

Berdasarkan tabel diatas durasi waktu pengujian tanpa kakas menghabiskan rata-rata waktu 450,75 detik.

Tabel 6. Waktu Menghasilkan Kasus Uji Dengan Kakas

No	Penguji	Waktu Pengujian
1	Penguji 1	55 detik
2	Penguji 2	103 detik
3	Penguji 3	74 detik
4	Penguji 4	73 detik

Perhitungan waktu pada pengujian dengan kakas, semua penguji menyelesaikan dengan waktu rata-rata 76,25 detik. Durasi waktu pengujian kedua ini jauh lebih cepat dibanding tanpa kakas pada pengujian kedua. Berikut grafik perbandingan waktu pengujian kedua.



Gambar 7. Perbandingan Waktu Menghasilkan Kasus Uji Tanpa Kakas dan Dengan Kakas

Dari hasil perbandingan waktu pengujian *pairwise testing* dengan kakas dan tanpa kakas pada ukuran 4x4x5x4 dapat dilihat pada gambar 5. Jadi dapat dihitung persentase perbandingan setiap penguji dengan menghitung waktu pengujian tanpa kakas dibagi waktu pengujian dengan kakas lalu dikali 100 setiap pengujinya. Berikut hasil perhitungan persentasenya

Tabel 8. Persentase Saat Menghasilkan Kasus Uji Tanpa Kakas dan Dengan Kakas ukuran 4x4x4x5

No	Penguji	Persentase perbandingan
1	Penguji 1	614,5%
2	Penguji 2	588,3%
3	Penguji 3	694,5%
4	Penguji 4	370,9%

4.2 Analisis Hasil Pengujian

RQ1. Apakah kakas yang dibangun sudah tepat?

Pengujian kakas dilakukan dengan 2 ukuran kombinasi data studi kasus yang berbeda, dengan 2 ukuran kombinasi yang berbeda, dapat mengetahui bahwa ada perbedaan waktu yang cukup jauh jika menggunakan ukuran 4x4x4x5 dibanding 2x2x3. Durasi waktu saat melakukan pengujian perangkat lunak berpengaruh dalam proses pengujian[10] Perbedaan waktu antara pengujian untuk menghasilkan kasus uji dengan kakas dan tanpa kakas jauh berbeda pada kedua ukuran, celah waktu pada ukuran 2x2x3 lebih sedikit dibanding dengan 4x4x4x5. Celah waktu setiap ukuran dapat dilihat pada tabel 4 dan 7. Tabel tersebut menghitung celah waktu dengan hasil persentase, jika dilihat persentase kedua tabel tersebut berbeda jauh. Jadi semakin besar ukuran, jarak antara menggunakan kakas dan tanpa menggunakan kakas lebih jauh celah waktunya,

Perbedaan celah waktu tidak hanya pada ukuran yang berbeda, tetapi setiap penguji juga mempunyai celah waktu berbeda dan waktu setiap pengujian tanpa kakas dan dengan kakas juga berbeda. Ada beberapa faktor yang menyebabkan perbedaan waktu tersebut. Pengalaman penguji menjadi salah satu faktor perbedaan hasil pengukuran waktu. Penguji 4 lebih unggul selama melakukan pengujian *pairwise testing*, dengan pengalamannya dalam bidang pengujian lebih dibanding penguji lainnya. Pengujian dilakukan oleh mahasiswa Universitas Telkom jurusan S1 Informatika dengan status mahasiswa tingkat 4 atau tingkat akhir, dan sudah pernah melakukan pengujian perangkat lunak saat kuliah maupun diluar kegiatan kuliah Status penguji ini penting untuk diketahui karena pengujian tidak bisa dilakukan dengan yang tidak pernah sama sekali melakukan pengujian perangkat lunak.

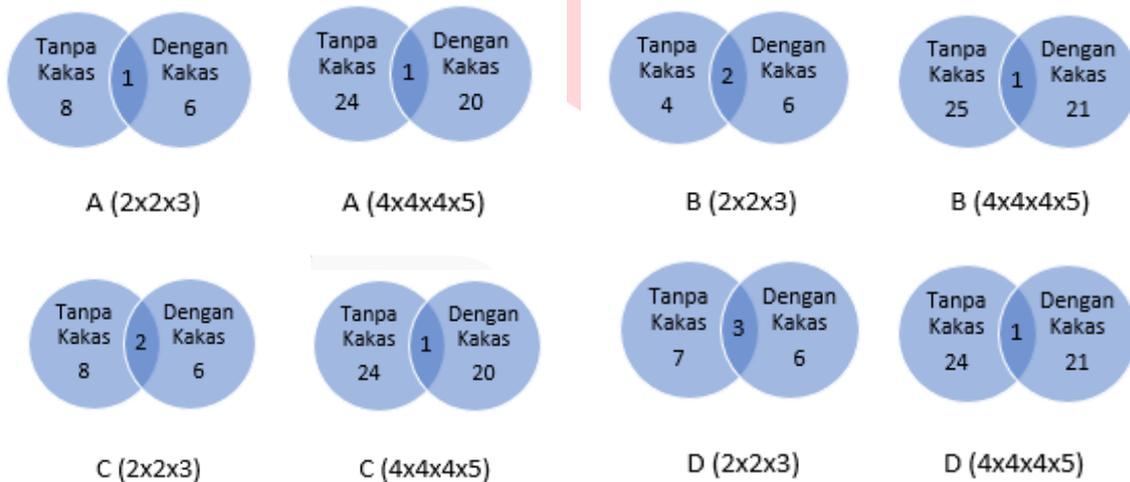
Pengujian dengan menggunakan kakas terbantu oleh Orthogonal Array yang sudah memetakan kemungkinan uji kasus yang akan dihasilkan. Seperti contoh Orthogonal Array gambar 1, ketika uji kasus dimasukkan ke kakas dan ditekan tombol generate akan langsung mengeluarkan hasil *pairwise* nya. Sistem tersebut yang dapat mempercepat hasil kasus uji dengan kakas *pairwise testing*. Perangkat lunak akan selalu perlu dicoba dan dimonitor [11]. Oleh karena itu, kakas untuk menghasilkan kasus uji ini dinilai tepat karena dapat membantu pengujian menghasilkan kasus uji dengan lebih cepat dan tepat.

RQ2. Bagaimana hasil kasus uji dengan menggunakan kakas *Pairwise Testing* dari hasil pengujian?

Dari hasil pengujian *Pairwise Testing* dengan atau tanpa kakas perlu dicocokkan dengan hasil Orthogonal Array, karena hasil dari pengujian *Pairwise Testing* menggunakan Orthogonal Array untuk mendapatkan kasus uji. Perlu dibuat diagram venn untuk mengetahui banyaknya data yang sama atau berkesinambungan pada pengujian dengan kakas dan tanpa kakas [9]

Berdasarkan diagram venn pada gambar 6 terdapat jumlah kasus uji yang dihasilkan pengujian 1 sampai 4. Terlihat bahwa jumlah kasus uji yang sama atau yang beririsan tidak lebih dari 3 kasus uji. Diagram venn A dan D menunjukkan hasil (hasil kasus uji ada pada lampiran) dari pengujian 1 dan 4. Ketika dicocokkan dengan Orthogonal Array, hasil pengujian tanpa kakas dan dengan kakas pada kedua ukuran memiliki kecocokan dengan Orthogonal Array, artinya kedua pengujian tersebut dapat melakukan pengujian dengan manual dan dengan kakas. Tetapi pengujian dengan kakas lebih cepat waktunya dan dapat dikatakan bermanfaat untuk ukuran yang lebih banyak kombinasinya (4x4x4x5).

Sedangkan pada diagram venn B dan C hasil pengujian 2 dan 3 tanpa kakas hasilnya tidak sesuai dengan Orthogonal Array, artinya untuk pengujian 2 dan 3 lebih cocok untuk menggunakan kakas saat melakukan pengujian *pairwise testing*. Setelah mengetahui bahwa 2 dari 4 pengujian bisa melakukan pengujian *pairwise testing* tanpa kakas, maka dapat dilihat kembali dari pengalaman pengujian, bahwa pengujian 1 dan 4 terlihat lebih banyak pengalaman dalam pengujian perangkat lunak. Berikut adalah gambar diagram venn untuk hasil pengujian *pairwise testing* penelitian ini.



Gambar 8. Diagram Venn Jumlah Hasil Kasus Uji Dengan Metode *Pairwise Testing*

Keterangan:

- A = Pengujian 1
- B = Pengujian 2
- C = Pengujian 3
- D = Pengujian 4

Berdasarkan diagram venn **gambar 6** dapat disimpulkan hasil kakas yang tidak tercakup oleh pengujian manual, hasil kakas yang tercakup oleh pengujian manual, dan seberapa banyak kesalahan kakas yang tercakup oleh pengujian manual. Pada diagram venn diatas merupakan hasil dari data masukan yang sama antara menggunakan kakas dan tidak menggunakan kakas. Hasil pencocokan dengan Orthogonal Array pada pengujian tanpa kakas dan dengan kakas ternyata banyak hasil kasus uji tanpa kakas yang tidak cocok dengan Orthogonal Array dan hanya bagian yang beririsan dengan kakas yang cocok dengan Orthogonal Array.

5. Kesimpulan

Berdasarkan hasil yang telah disampaikan dengan tujuan membuat dan menganalisis kakas untuk dapat menghasilkan kasus uji dengan metode *Pairwise Testing* dalam pengujian perangkat lunak dapat disimpulkan, bahwa menghasilkan kasus uji dengan *Pairwise Testing* dilihat dari segi waktu dan hasil dari seluruh pengujian lebih cocok

menggunakan kakas, tetapi jika dari sisi hasil lebih banyak tanpa menggunakan kakas walaupun hanya 2 penguji saja yang hasilnya cocok dengan Orthogonal Array.

Pengalaman penguji pada penelitian ini berpengaruh pada pengujian *Pairwise Testing*, ini terbukti ketika hasil kasus uji dicocokkan dengan Orthogonal Array bahwa penguji 1 dan 4 yang memiliki pengalaman lebih dan hasil kasus uji tanpa kakas lebih banyak cocok. Kasus uji yang dihasilkan dengan menggunakan kakas pairwise testing memiliki peluang untuk menemukan cacat pada perangkat lunak akan tetapi dapat mereduksi kasus uji yang dihasilkan. Maka saran untuk penelitian selanjutnya adalah bagaimana jika penguji ditambah dengan pengalaman penguji yang beragam apakah akan menambah kecocokan hasil tanpa kakas yang lebih akurat.

Daftar Pustaka

- [1] Paternostro, L. M. A., & Slavescu, M. (2007). Formal Test Case Definitions *U.S. Patent No. 7,296,188*. Washington, DC: U.S. Patent and Trademark Office.
- [2] Harrold, M. J. (2000, May). Testing: a roadmap. In *Proceedings of the Conference on the Future of Software Engineering* (pp. 61-72).
- [3] Salamah, S., Ochoa, O., & Jacquez, Y. (2015, January). Using pairwise testing to verify automatically-generated formal specifications. In *2015 IEEE 16th International Symposium on High Assurance Systems Engineering* (pp. 279-280). IEEE.
- [4] Czerwonka, J. (2008). Pairwise testing in the real world: Practical extensions to test-case scenarios. *Microsoft Corporation, Software Testing Technical Articles*.
- [5] Shi, L., Nie, C., & Xu, B. (2005, May). A software debugging method based on pairwise testing. In *International Conference on Computational Science* (pp. 1088-1091). Springer, Berlin, Heidelberg.
- [6] Kuhn, D. R., Kacker, R., & Lei, Y. (2008). Automated combinatorial test methods: Beyond pairwise testing. *Crosstalk, Journal of Defense Software Engineering*, 21(6), 22-26.
- [7] Kacker, R. N., Lagergren, E. S., & Filliben, J. J. (1991). Taguchi's orthogonal arrays are classical designs of experiments. *Journal of research of the National Institute of Standards and Technology*, 96(5), 577.
- [8] Itkonen, J., & Mäntylä, M. V. (2014). Are test cases needed? Replicated comparison between exploratory and test-case-based software testing. *Empirical Software Engineering*, 19(2), 303-342.
- [9] Shade, A., & Handelsman, J. (2012). Beyond the Venn diagram: the hunt for a core microbiome. *Environmental microbiology*, 14(1), 4-12.
- [10] Xie, M., & Yang, B. (2003). A study of the effect of imperfect debugging on software development cost. *IEEE Transactions on Software Engineering*, 29(5), 471-473.
- [11] Bertolino, A. (2007, May). Software testing research: Achievements, challenges, dreams. In *Future of Software Engineering (FOSE'07)* (pp. 85-103). IEEE.

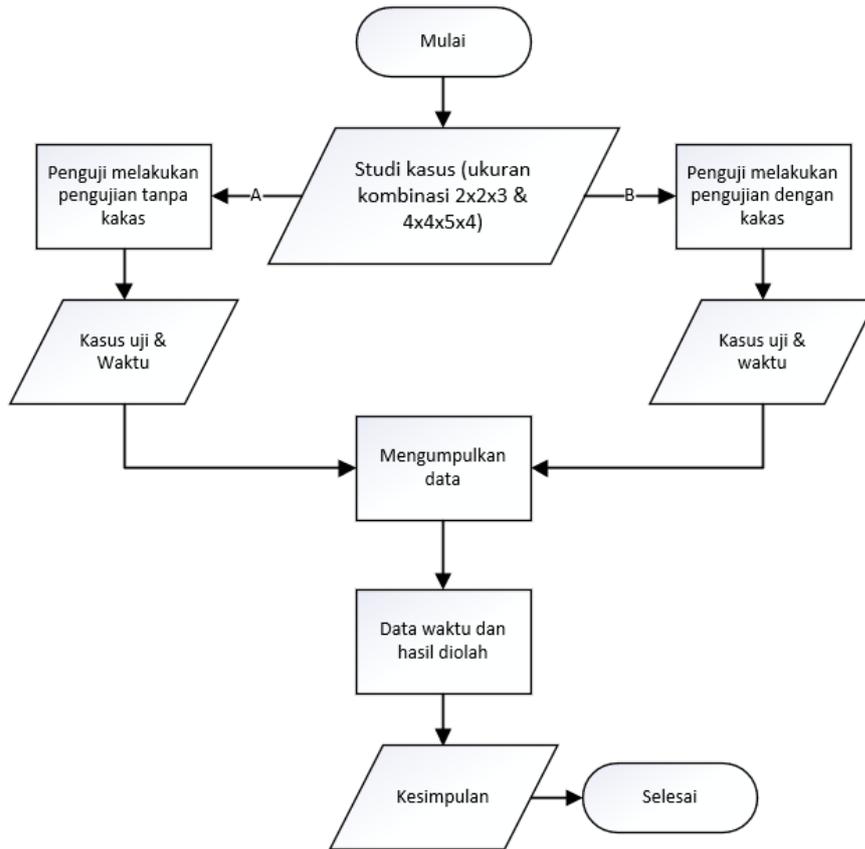
Lampiran

1. Studi kasus sudah pernah digunakan pada paper yang berjudul:

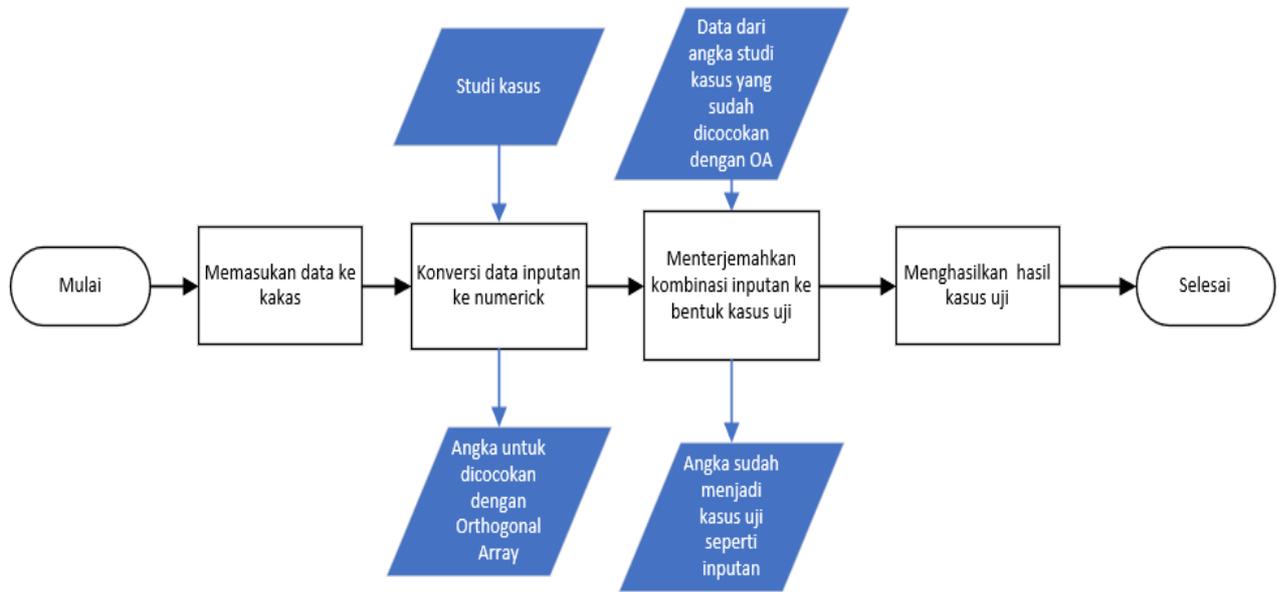
- 1.1 Development and evaluation of HAYST metode tool
- 1.2 Introduction to combination testing using orthogonal array, software test press/gijutsu-hyohron vol 2 pages 89-107, 2006

2. Skema Evaluasi *Pairwise Testing*

Berikut merupakan skema evaluasi yang dilakukan dalam penelitian ini.



Skema evaluasi *pairwise testing*



3. Hasil pengujian dari penguji

Orthogonal Array:

Run	Columns			
	1	2	3	4
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

Penguji 1 :

size	style	effect
	8 bold	shade
	8 italic	upper
~8	~bold	~shade
	9 italic	shade
	9 bold	upper
~9	~italic	~upper
	10 bold	shade
	10 italic	upper

size	style	effect	color
8	normal	underline	blue
8	bold	shade	green
8	italic	emboss	yellow
8	bold&italic	upper	red
~8	~normal	~underline	~blue
9	normal	upper	blue
9	bold	underline	green
9	italic	shade	yellow
9	bold&italic	emboss	red
~9	~bold	~shade	~green
10	normal	emboss	blue
10	bold	upper	green
10	italic	underline	yellow
10	bold&italic	shade	red
~10	~italic	~upper	~yellow
12	normal	shade	blue
12	bold	emboss	green
12	italic	upper	yellow
12	bold&italic	underline	red
~12	~bold&italic	~upper	~red
14	normal	underline	blue
14	bold	shade	green
14	italic	emboss	yellow
14	bold&italic	upper	red

Penguji 2:

size	style	effect
32	italic	shade
14	bold	shade
20	bold	shade
32	bold	shade

color	size	style	effect
blue	18	italic	underline
blue	20	italic	underline
blue	28	italic	underline
blue	8	italic	underline
blue	10	italic	underline
blue	18	bold&italic	underline
blue	20	bold&italic	underline
blue	28	bold&italic	underline
blue	8	bold&italic	underline
blue	10	bold&italic	underline
blue	18	normal	shade
blue	20	normal	shade
blue	28	normal	shade
blue	8	normal	shade
blue	10	normal	shade
blue	18	bold	shade
blue	20	bold	shade
blue	28	bold	shade
blue	8	bold	shade
blue	10	bold	shade
blue	18	italic	shade
blue	20	italic	shade
blue	28	italic	shade
blue	8	italic	shade
blue	10	italic	shade

Penguji 3:

size	effect	style
8	shade	normal
8	shade	normal
8	shade	normal
24	underline	bold
24	underline	bold
24	underline	normal
80	shade	normal
80	shade	bold

size	effect	style	color
8	underline	normal	blue
8	underline	normal	blue
8	underline	normal	blue
8	shade	normal	blue
8	shade	normal	blue
8	underline	bold	green
8	underline	bold	green
8	underline	bold	green
9	shade	bold	green
9	shade	bold	green
9	shade	bold	green
9	shade	bold	green
9	shade	bold	green
9	shade	bold	green
9	shade	bold	green
10	embos	italic	yellow
10	embos	italic	yellow
10	embos	italic	yellow
12	embos	italic	yellow
12	embos	italic	yellow
12	embos	italic	yellow
16	underline	bold&italic	red
16	underline	bold&italic	red
16	underline	bold&italic	red

Penguji 4:

Size	effect	style
8	underline	bold
8	shade	italic
9	underline	bold
9	shade	italic
10	underline	bold
10	shade	italic
10	shade	bold

Size	effect	style	color
8	underline	normal	blue
8	shade	bold	green
8	emboss	italin	yellow
8	upper	bold&italic	red
8	underline	bold	green
9	underline	normal	red
9	shade	bold	yellow
9	emboss	italic	blue
9	upper	bold&italic	green
9	shade	italic	yellow
10	underline	bold	green
10	shade	normal	blue
10	emboss	italic	yellow
10	upper	bold&italic	red
10	emboss	normal	green
12	underline	normal	yellow
12	shade	italic	red
12	emboss	bold	green
12	upper	bold&italic	blue
12	emboss	italic	blue
14	underline	bold&italic	blue
14	shade	normal	yellow
14	emboss	bold&italic	red
14	upper	italic	green