

# Optimasi Pengujian Kombinatorial dengan T-Way Testing untuk Validasi Perangkat Lunak Berbasis Web

Andrew Isra Saputra DB - 13523110<sup>1</sup>

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

[andrew.israsaputra@gmail.com](mailto:andrew.israsaputra@gmail.com), [13523110@std.stei.itb.ac.id](mailto:13523110@std.stei.itb.ac.id)

**Abstrak**— Pengujian perangkat lunak berbasis web sering kali melibatkan sejumlah besar parameter dan kombinasi nilai yang harus diuji, yang dapat menjadi sangat kompleks dan memakan waktu. Untuk mengatasi tantangan ini, pendekatan T-Way Testing diperkenalkan sebagai solusi untuk mengurangi jumlah kombinasi uji yang perlu dijalankan, dengan tetap memastikan cakupan pengujian yang memadai terhadap interaksi antar parameter. Dalam pengujian T-Way, hanya interaksi antar T parameter yang diuji, sehingga dapat mengurangi jumlah kasus uji secara signifikan.

Makalah ini membahas optimasi pengujian perangkat lunak berbasis web dengan menggunakan T-Way Testing, dengan fokus pada alat penghasil kombinasi uji yang efisien, seperti algoritma IPOG (In-Parameter Order Generalization). Melalui penerapan algoritma ini, makalah ini bertujuan untuk menunjukkan bagaimana pengujian dapat disederhanakan tanpa mengorbankan kualitas dan cakupan pengujian. Dengan mengurangi jumlah kombinasi uji yang diperlukan, waktu dan biaya pengujian dapat diminimalkan, sementara tetap memastikan bahwa interaksi antar parameter diuji secara menyeluruh. Hasil dari penerapan pendekatan ini menunjukkan bahwa penggunaan T-Way Testing dalam pengujian perangkat lunak berbasis web tidak hanya mengurangi waktu dan biaya, tetapi juga meningkatkan efektivitas pengujian secara keseluruhan.

**Kata Kunci**—T-Way Testing, Pengujian Perangkat Lunak, IPOG, Kombinasi Uji, Pengujian Perangkat Lunak Berbasis Web.

## I. PENDAHULUAN

Dalam era digital saat ini, perangkat lunak berbasis web memainkan peran vital dalam berbagai aspek kehidupan, mulai dari layanan perbankan hingga pendidikan. Kompleksitas dan keragaman fungsi yang ditawarkan oleh aplikasi web menuntut proses pengujian yang efektif untuk memastikan kualitas dan keandalannya. Salah satu tantangan utama dalam pengujian perangkat lunak adalah banyaknya kombinasi input dan konfigurasi yang mungkin terjadi, yang membuat pengujian secara menyeluruh menjadi tidak praktis.

Pengujian kombinatorial muncul sebagai solusi untuk mengatasi tantangan ini dengan mengurangi jumlah kasus uji tanpa mengorbankan cakupan pengujian. Metode ini fokus pada pengujian semua kemungkinan kombinasi dari sejumlah parameter tertentu, yang dikenal dengan istilah t-way testing. Pendekatan ini memungkinkan deteksi cacat yang mungkin

tidak teridentifikasi melalui metode pengujian tradisional. Menurut Ahmed et al., penggunaan algoritma cuckoo search dalam pengujian kombinatorial dapat menghasilkan suite uji yang lebih efisien untuk pengujian perangkat lunak berbasis konfigurasi [1].

Selain itu, Zamli et al. mengembangkan strategi hibrida yang menggabungkan Q-learning dan algoritma sine-cosine untuk memecahkan masalah minimisasi suite uji kombinatorial. Pendekatan ini berhasil mengurangi ukuran suite uji secara signifikan, yang berdampak pada efisiensi proses pengujian perangkat lunak [2]. Dalam konteks pengujian perangkat lunak berbasis web, metode t-way testing menawarkan pendekatan yang sistematis untuk mengidentifikasi kombinasi input yang kritis. Dengan demikian, pengujian dapat difokuskan pada area yang paling rentan terhadap kesalahan, sehingga meningkatkan keandalan dan kualitas aplikasi web secara keseluruhan.

Penerapan metode optimasi dalam pengujian perangkat lunak menunjukkan bahwa penggunaan teknik seperti Ant Colony Optimization (ACO) dan algoritma genetika dapat membantu menyelesaikan masalah optimasi kombinatorial dalam konteks pengujian perangkat lunak. Misalnya, ACO telah diterapkan dalam berbagai masalah optimasi, termasuk dalam pengujian perangkat lunak, untuk mencari solusi optimal dengan meniru perilaku koloni semut dalam mencari jalur terpendek [3]. Dengan demikian, integrasi metode t-way testing tidak hanya meningkatkan efisiensi pengujian tetapi juga memastikan bahwa aplikasi yang dikembangkan memenuhi standar kualitas yang tinggi.

## II. LANDASAN TEORI

### 1. Pengujian Perangkat Lunak Berbasis Web

Pengujian perangkat lunak berbasis web melibatkan pengujian aplikasi yang diakses melalui web browser. Aplikasi web sering kali bersifat dinamis, interaktif, dan dipengaruhi oleh banyak faktor eksternal, seperti sistem operasi, perangkat keras, koneksi internet, dan browser yang digunakan oleh pengguna. Karakteristik ini membuat pengujian perangkat lunak berbasis web lebih kompleks dibandingkan perangkat lunak desktop tradisional.

### a. Kompleksitas Pengujian Aplikasi Web

Perangkat lunak berbasis web sering kali berinteraksi dengan berbagai teknologi yang dapat berfungsi secara berbeda di berbagai platform. Misalnya, aplikasi web harus kompatibel dengan berbagai jenis perangkat keras (misalnya, komputer desktop, ponsel pintar), berbagai browser (seperti Chrome, Firefox, Safari), serta sistem operasi yang berbeda (Windows, macOS, Linux). Setiap kombinasi ini bisa mempengaruhi cara aplikasi bekerja. Selain itu, aplikasi berbasis web harus mampu menangani berbagai tipe input, pengaturan konfigurasi server, serta kondisi jaringan yang berbeda. Oleh karena itu, pengujian harus memperhitungkan berbagai skenario pengguna dan konfigurasi teknis yang berbeda.

### b. Keandalan dan Kualitas Perangkat Lunak Berbasis Web

Keandalan perangkat lunak sangat penting dalam aplikasi berbasis web karena kesalahan atau bug dapat menyebabkan gangguan pada banyak pengguna yang mengakses aplikasi secara bersamaan. Oleh karena itu, pengujian perangkat lunak bertujuan untuk mengidentifikasi dan memperbaiki cacat yang dapat mengurangi pengalaman pengguna dan memengaruhi performa aplikasi. Selain itu, penting untuk memastikan bahwa aplikasi dapat berfungsi dengan baik di berbagai perangkat dan browser yang digunakan oleh pengguna di seluruh dunia.

### c. Pengujian untuk Memastikan Fungsi dan Interaksi

Pengujian perangkat lunak berbasis web bertujuan untuk memastikan bahwa aplikasi tidak hanya berfungsi dengan baik di berbagai platform, tetapi juga dapat menangani berbagai input dan interaksi dengan sistem lain (misalnya, server, database). Proses pengujian mencakup pengujian fungsionalitas (seperti klik tombol, pengisian formulir, dan pengunduhan file) dan pengujian non-fungsionalitas (seperti performa aplikasi, keamanan, dan kemampuan skalabilitas).

## 2. Pengujian Kombinatorial

Pengujian kombinatorial adalah pendekatan untuk memilih subset dari semua kemungkinan kombinasi parameter masukan untuk mengurangi jumlah pengujian yang perlu dilakukan, sambil tetap menjaga efektivitas dalam mendeteksi cacat perangkat lunak. Konsep pengujian kombinatorial pertama kali diperkenalkan oleh Ray Bohlin pada tahun 1995 untuk mengatasi masalah kompleksitas dalam pengujian perangkat lunak.

### a. Definisi dan Tujuan Pengujian Kombinatorial

Pengujian kombinatorial bertujuan untuk memastikan bahwa kombinasi yang relevan dari parameter input diuji. Pengujian ini berdasarkan pada prinsip bahwa sebagian besar cacat perangkat lunak disebabkan oleh interaksi antar dua atau tiga parameter. Oleh karena itu, pengujian yang mencakup kombinasi beberapa parameter secara efisien dapat mendeteksi banyak cacat tanpa perlu menguji semua kemungkinan kombinasi secara eksplisit.

### b. Prinsip Pairwise Testing

Pairwise testing adalah jenis pengujian kombinatorial yang berfokus pada pengujian semua pasangan kombinasi dua

parameter. Penelitian menunjukkan bahwa sebagian besar cacat perangkat lunak dapat ditemukan dengan menguji kombinasi dua parameter. Ini mengurangi jumlah kasus uji secara signifikan sambil tetap memberikan cakupan pengujian yang baik. Misalnya, jika ada 5 parameter, dan masing-masing parameter memiliki 4 nilai yang mungkin, maka total kombinasi yang harus diuji adalah  $4^5 = 1024$  kasus. Namun, dengan pairwise testing, jumlah kasus dapat dikurangi menjadi hanya 16 kombinasi.

### c. Pengujian Kombinatorial untuk Kombinasi Lebih dari Dua Parameter

Meskipun pengujian pairwise sangat efektif, banyak masalah perangkat lunak muncul akibat interaksi antara lebih dari dua parameter. Untuk itu, pengujian kombinatorial diperluas untuk mencakup lebih banyak parameter (seperti tiga atau empat parameter sekaligus). Pendekatan ini dikenal sebagai pengujian T-Way, di mana "T" merujuk pada jumlah parameter yang diuji dalam satu kombinasi. Tujuan dari pengujian ini adalah untuk memastikan bahwa setiap kombinasi interaksi antara T parameter diuji.

## 3. T-Way Testing

T-Way testing adalah sebuah teknik dalam pengujian kombinatorial yang menguji semua kombinasi interaksi hingga T parameter. Sebagai contoh, pada 2-Way testing (*pairwise testing*), hanya interaksi dua parameter yang diuji, sedangkan pada 3-Way testing, pengujian dilakukan pada semua kombinasi tiga parameter, dan seterusnya.

### a. Definisi dan Tujuan T-Way Testing

T-Way testing bertujuan untuk menguji kombinasi interaksi antar parameter dalam perangkat lunak dengan mempertimbangkan interaksi yang lebih kompleks daripada hanya dua parameter. Dalam T-Way testing, jumlah parameter yang diuji dapat bervariasi, mulai dari dua hingga beberapa parameter (misalnya, 3-Way testing atau 4-Way testing). Teknik ini sangat berguna untuk menemukan bug yang muncul hanya ketika beberapa parameter berinteraksi, yang sering kali tidak terdeteksi dengan metode pengujian lainnya.

### b. Keuntungan T-Way Testing

T-Way testing mengoptimalkan pengujian dengan mengurangi jumlah kombinasi yang harus diuji, tetapi tetap dapat mencakup seluruh kemungkinan interaksi parameter yang penting. Dengan menggunakan teknik ini, pengujian dapat dilakukan dengan lebih efisien, karena hanya kombinasi parameter yang relevan yang diuji. Selain itu, pengujian ini mengurangi waktu dan biaya pengujian, yang sangat penting dalam pengujian perangkat lunak berbasis web yang kompleks.

### c. Algoritma dalam T-Way Testing

Beberapa algoritma yang digunakan dalam T-Way testing untuk menghasilkan suite uji yang efisien antara lain:

- 1) IPOG (In-Parameter-Order General): IPOG adalah algoritma yang menghasilkan kombinasi parameter secara iteratif berdasarkan urutan parameter.

Algoritma ini digunakan untuk menghasilkan kombinasi minimal dalam pengujian.

- 2) AETG (Automatic Efficient Test Generator): AETG menggunakan algoritma berbasis heuristik yang dirancang untuk menghasilkan suite uji yang efisien dengan mencakup semua kombinasi interaksi hingga T parameter.
- 3) Greedy Algorithm: Algoritma ini memilih kombinasi parameter yang paling mungkin mengungkapkan cacat berdasarkan informasi yang tersedia.
- 4) Genetic Algorithm (GA): Menggunakan prinsip seleksi alam untuk mengoptimalkan kombinasi kasus uji. GA diterapkan untuk memilih solusi terbaik secara iteratif, berdasarkan pemilihan, perkawinan, dan mutasi dari individu (kandidat solusi).

#### d. Penerapan T-Way Testing dalam Pengujian Perangkat Lunak Berbasis Web

Penerapan T-Way testing pada perangkat lunak berbasis web sangat bermanfaat untuk memeriksa kombinasi yang lebih kompleks, yang mungkin terjadi akibat interaksi antara berbagai faktor, seperti browser yang berbeda, sistem operasi, dan pengaturan konfigurasi lainnya. T-Way testing memastikan bahwa pengujian perangkat lunak berbasis web dapat mencakup semua kombinasi yang relevan, tanpa harus menguji semua kemungkinan secara eksplisit, yang bisa sangat memakan waktu dan biaya.

### III. ANALISIS DAN PEMBAHASAN

#### A. Desain Pengujian

##### 1. Identifikasi Parameter Pengujian

Pada tahap awal, pengujian dimulai dengan identifikasi parameter-parameter yang berperan penting dalam pengoperasian perangkat lunak berbasis web. Parameter-parameter ini mencakup elemen-elemen yang akan diuji untuk memastikan bahwa aplikasi berfungsi sesuai dengan ekspektasi di berbagai konfigurasi dan kondisi. Parameter yang relevan dalam pengujian aplikasi web dapat meliputi:

- 1) Browser: Misalnya, Chrome, Firefox, Safari, dan Edge. Setiap browser dapat mempengaruhi rendering dan kinerja aplikasi secara berbeda.
- 2) Sistem Operasi: Windows, macOS, dan Linux. Aplikasi berbasis web harus berfungsi dengan baik di berbagai sistem operasi.
- 3) Resolusi Layar: Aplikasi harus responsif dan menyesuaikan dengan berbagai resolusi layar seperti 1024x768, 1366x768, dan 1920x1080.
- 4) Koneksi Jaringan: Wi-Fi, 4G, dan 5G, yang bisa mempengaruhi kinerja aplikasi dalam kondisi jaringan yang berbeda.

##### 2. Penyusunan Kombinasi Uji dengan T-Way Testing

Setelah parameter-parameter yang relevan teridentifikasi, langkah berikutnya adalah penyusunan kombinasi pengujian. Dalam pengujian tradisional, setiap kombinasi parameter diuji satu per satu. Namun, dalam pengujian menggunakan T-Way

testing, hanya kombinasi parameter yang melibatkan interaksi antara dua atau lebih parameter yang diuji.

Misalnya, dalam 2-Way testing, semua kombinasi interaksi antara dua parameter yang berbeda diuji, sedangkan 3-Way testing memperhitungkan interaksi antara tiga parameter. Dengan T-Way testing, tujuan utamanya adalah untuk meminimalkan jumlah kombinasi pengujian tanpa mengorbankan cakupan interaksi antar parameter.

Sebagai contoh, untuk 2-Way testing, jika ada 4 parameter yang diuji, dan masing-masing parameter memiliki 3 nilai, maka jumlah total kombinasi yang akan diuji menggunakan pengujian lengkap adalah  $3^4 = 81$  kombinasi. Namun, dengan menggunakan 2-Way testing, jumlah kombinasi dapat dikurangi secara signifikan, tergantung pada strategi yang digunakan.

#### 3. Alat Penghasil Kombinasi Uji (Test Case Generator)

Untuk memfasilitasi penerapan T-Way testing, alat bantu seperti IPOG (In-Parameter Order Generalization) digunakan. IPOG adalah algoritma yang dirancang untuk mengurangi jumlah kombinasi pengujian yang dibutuhkan dengan tetap mempertahankan cakupan yang diperlukan. Alat ini menghasilkan kombinasi parameter yang memenuhi syarat T-Way testing dan mengurangi redundant test case yang sering ditemukan dalam pendekatan pengujian lengkap.

Prinsip kerja IPOG adalah dengan mengurutkan nilai-nilai dari setiap parameter secara sistematis, kemudian menghasilkan kombinasi uji yang mencakup interaksi antar parameter sesuai dengan tingkat T-Way yang diinginkan. Misalnya, dalam pengujian 2-Way (atau pasangan nilai antar dua parameter), IPOG akan menghasilkan kombinasi uji yang mencakup semua pasangan nilai yang mungkin, tanpa mengulang kombinasi yang sudah diuji sebelumnya. Dengan demikian, jumlah kasus uji dapat diminimalkan tanpa mengabaikan cakupan yang diperlukan.

#### B. Implementasi Pengujian

Setelah desain pengujian selesai, langkah berikutnya adalah mengimplementasikan pengujian tersebut pada perangkat lunak berbasis web yang sedang diuji. Proses implementasi mencakup pemilihan alat, penerapan teknik pengujian, dan eksekusi pengujian pada aplikasi. Berikut adalah langkah-langkah rinci yang dilakukan dalam implementasi pengujian dengan T-Way Testing.

##### 1. Pemilihan Alat Pengujian

Pemilihan alat yang tepat merupakan langkah penting dalam pengujian perangkat lunak berbasis web. Beberapa alat yang dapat digunakan untuk implementasi pengujian kombinatorial dan T-Way Testing adalah:

- 1) ActGen, adalah alat yang dapat menghasilkan kombinasi uji untuk pengujian kombinatorial, termasuk T-Way Testing. Alat ini memungkinkan pengguna untuk menentukan jumlah parameter yang akan diuji serta nilai-nilai yang relevan dari masing-masing parameter tersebut.
- 2) TConfig, adalah alat yang digunakan untuk mengkonfigurasi dan menghasilkan kasus uji berdasarkan T-Way Testing. Alat ini dapat

diintegrasikan dengan berbagai platform pengujian perangkat lunak.

- 3) IPOG, adalah algoritma yang digunakan untuk mengoptimalkan kombinasi uji dan mengurangi jumlah kasus uji yang diperlukan tanpa mengurangi efektivitas pengujian. IPOG mengurangi redundansi dalam kasus uji dan memastikan semua interaksi antarparameter diuji.
- 4) Selenium, adalah alat otomatisasi pengujian yang digunakan untuk menjalankan skrip pengujian pada aplikasi berbasis web. Selenium mendukung berbagai browser dan memungkinkan pengujian dilakukan secara otomatis, sehingga pengujian dapat dilakukan lebih cepat dan lebih efisien.
- 5) JUnit, adalah alat pengujian berbasis Java yang digunakan untuk mengimplementasikan dan menjalankan pengujian unit dalam pengembangan perangkat lunak. Dengan menggunakan JUnit, kita dapat mengotomatiskan pengujian untuk memastikan bahwa semua kasus uji berhasil dijalankan.

## 2. Pengaturan Lingkungan Pengujian

Sebelum implementasi pengujian, penting untuk memastikan bahwa lingkungan pengujian sudah terkonfigurasi dengan baik. Hal ini mencakup pengaturan perangkat keras, perangkat lunak, dan alat pengujian yang diperlukan.

### 1) Perangkat Keras

Sistem pengujian harus memiliki spesifikasi yang memadai untuk menjalankan aplikasi berbasis web serta alat pengujian secara bersamaan tanpa hambatan kinerja. Misalnya, prosesor yang cepat, memori yang cukup besar, dan penyimpanan yang memadai.

### 2) Perangkat Lunak

Pastikan bahwa semua perangkat lunak yang diperlukan untuk pengujian terinstal dengan benar. Ini termasuk server web untuk menjalankan aplikasi, alat otomatisasi pengujian seperti Selenium, dan alat untuk menghasilkan kasus uji seperti TConfig atau ActGen.

### 3) Database

Jika aplikasi berbasis web melibatkan interaksi dengan database, pastikan bahwa database yang digunakan dalam pengujian adalah versi yang sama dengan yang digunakan dalam lingkungan produksi. Ini akan memastikan bahwa pengujian mencerminkan keadaan aplikasi yang sebenarnya.

## 3. Pengujian

Dalam konteks pengujian perangkat lunak berbasis web, dipilih empat parameter penting yang dapat memengaruhi fungsionalitas dan kinerja sistem. Keempat parameter yang dipilih adalah Sistem Operasi, Browser, Resolusi Layar, dan Kecepatan Koneksi Internet. Pemilihan parameter ini berdasarkan pada pengaruh langsung mereka terhadap interaksi pengguna dengan aplikasi web.

- 1) Sistem Operasi: Perangkat lunak berbasis web harus dapat berfungsi dengan baik di berbagai sistem operasi yang digunakan oleh pengguna. Sistem

operasi yang diuji mencakup Windows, Linux, dan macOS

- 2) Browser: Kompatibilitas antar browser juga menjadi faktor krusial dalam pengujian aplikasi web. Browser yang diuji dalam penelitian ini adalah Chrome, Firefox, dan Safari
- 3) Resolusi Layar: Responsivitas aplikasi web terhadap berbagai ukuran layar perlu diuji untuk memastikan aplikasi berfungsi dengan baik pada perangkat yang berbeda. Resolusi layar yang diuji meliputi 1024×768, 1366×768, dan 1920×1080
- 4) Kecepatan Koneksi Internet: Pengalaman pengguna aplikasi web sangat dipengaruhi oleh kecepatan koneksi internet. Oleh karena itu, parameter ini diuji dengan kondisi yang bervariasi, meliputi 3G, 4G, dan Wi-Fi

Matriks berikut menunjukkan kombinasi 4 parameter yang diuji dalam pengujian perangkat lunak berbasis web. Setiap baris menunjukkan kombinasi satu set nilai dari keempat parameter yang diuji, yang mencakup semua interaksi antar parameter untuk memastikan pengujian menyeluruh.

**Tabel 1.** Matriks Uji Kasus 4 Parameter

Sistem Operasi	Browser	Resolusi Layar	Kecepatan Koneksi
Windows	Chrome	1024x768	3G
Windows	Chrome	1366x768	4G
Windows	Chrome	1920x1080	Wi-Fi
Windows	Firefox	1366x768	4G
Windows	Firefox	1920x1080	3G
Windows	Firefox	1024x768	Wi-Fi
Windows	Safari	1920x1080	4G
Windows	Safari	1366x768	Wi-Fi
Windows	Safari	1024x768	3G
Linux	Chrome	1366x768	Wi-Fi
Linux	Chrome	1920x1080	4G
Linux	Chrome	1024x768	3G
Linux	Firefox	1920x1080	Wi-Fi
Linux	Firefox	1366x768	3G
Linux	Firefox	1024x768	4G
Linux	Safari	1366x768	4G
Linux	Safari	1920x1080	3G
Linux	Safari	1024x768	Wi-Fi
macOS	Chrome	1920x1080	4G

macOS	Chrome	1366x768	Wi-Fi
macOS	Chrome	1024x768	3G
macOS	Firefox	1366x768	3G
macOS	Firefox	1024x768	4G
macOS	Firefox	1920x1080	Wi-Fi
macOS	Safari	1024x768	4G
macOS	Safari	1366x768	3G
macOS	Safari	1920x1080	Wi-Fi

Selanjutnya dilakukan pengujian menggunakan selenium dengan script python.

```

1 from selenium import webdriver
2 from selenium.webdriver.common.by import By
3 from selenium.webdriver.common.keys import Keys
4 import time
5
6 # Matriks 4-Way Testing yang berisi kombinasi konfigurasi yang berbeda
7 combinations = [
8     ('Windows', 'Chrome', '1024x768', '3G'),
9     ('Windows', 'Chrome', '1366x768', '4G'),
10    ('Windows', 'Firefox', '1366x768', 'Wi-Fi'),
11    ('Linux', 'Chrome', '1920x1080', 'Wi-Fi'),
12    # Tambahkan kombinasi lainnya sesuai kebutuhan
13 ]
14
15 # Fungsi untuk menyesuaikan resolusi layar menggunakan Selenium
16 def set_resolution(driver, resolution):
17     if resolution == '1024x768':
18         driver.set_window_size(1024, 768)
19     elif resolution == '1366x768':
20         driver.set_window_size(1366, 768)
21     elif resolution == '1920x1080':
22         driver.set_window_size(1920, 1080)
23
24 # Fungsi untuk mengatur simulasi kecepatan koneksi menggunakan Chrome DevTools Protocol
25 def set_network_speed(driver, connection):
26     if connection == '3G':
27         driver.execute_cdp_cmd('Network.emulateNetworkConditions', {
28             'offline': False,
29             'latency': 5, # Latency dalam ms
30             'downloadThroughput': 500 * 1024, # Kecepatan download dalam bps (500 kbps)
31             'uploadThroughput': 500 * 1024 # Kecepatan upload dalam bps (500 kbps)
32         })
33     elif connection == '4G':
34         driver.execute_cdp_cmd('Network.emulateNetworkConditions', {
35             'offline': False,
36             'latency': 2, # Latency dalam ms
37             'downloadThroughput': 1500 * 1024, # Kecepatan download dalam bps (1500 kbps)
38             'uploadThroughput': 1500 * 1024 # Kecepatan upload dalam bps (1500 kbps)
39         })
40     elif connection == 'Wi-Fi':
41         driver.execute_cdp_cmd('Network.emulateNetworkConditions', {
42             'offline': False,
43             'latency': 1, # Latency dalam ms
44             'downloadThroughput': 5000 * 1024, # Kecepatan download dalam bps (5000 kbps)
45             'uploadThroughput': 5000 * 1024 # Kecepatan upload dalam bps (5000 kbps)
46         })

```

```

1 # Mulai pengujian untuk setiap kombinasi konfigurasi
2 for combination in combinations:
3     os, browser, resolution, connection = combination
4
5     # Tentukan WebDriver berdasarkan browser yang digunakan
6     if browser == 'Chrome':
7         driver = webdriver.Chrome(executable_path="path/to/chromedriver")
8     elif browser == 'Firefox':
9         driver = webdriver.Firefox(executable_path="path/to/geckodriver")
10
11    # Atur resolusi layar
12    set_resolution(driver, resolution)
13
14    # Atur kecepatan koneksi sesuai dengan konfigurasi
15    set_network_speed(driver, connection)
16
17    # Mulai mengakses halaman web yang ingin diuji
18    driver.get('https://your-web-application-url.com')
19
20    # Tunggu beberapa detik untuk memastikan halaman dimuat dengan baik
21    time.sleep(3)

```

```

23 # Periksa apakah halaman utama dimuat dengan benar
24 try:
25     assert "Home Page" in driver.title
26     print(f"Test passed for combination: {combination}")
27 except AssertionError:
28     print(f"Test failed for combination: {combination}")
29
30 # Tambahkan pengujian lain seperti interaksi dengan elemen di halaman
31 try:
32     # Contoh pengujian mengisi form pencarian
33     search_box = driver.find_element(By.NAME, "search")
34     search_box.send_keys("Test Query")
35     search_box.send_keys(Keys.RETURN)
36
37     # Tunggu hasil pencarian dimuat
38     time.sleep(3)
39
40     # Periksa hasil pencarian
41     results = driver.find_elements(By.CLASS_NAME, "search-result")
42     assert len(results) > 0
43     print(f"Search test passed for combination: {combination}")
44 except Exception as e:
45     print(f"Search test failed for combination: {combination}. Error: {e}")
46
47 # Tutup browser setelah pengujian selesai
48 driver.quit()

```

Gambar 1. Kode Python untuk Pengujian

1. Pengujian dilakukan untuk setiap kombinasi dalam matriks, dengan mengonfigurasi WebDriver sesuai dengan browser yang digunakan.
2. WebDriver yang sesuai diinisialisasi (Chrome, Firefox, Linux, dan MacOS), dan pengaturan resolusi layar serta kecepatan koneksi dijalankan untuk setiap kombinasi.
3. Setelah konfigurasi selesai, aplikasi web diakses menggunakan driver.get(). Halaman utama diuji untuk memastikan bahwa halaman utama dimuat dengan benar dengan memverifikasi judul halaman menggunakan assert.
4. Selanjutnya, pengujian dilakukan terhadap fitur pencarian di aplikasi web, di mana input teks dimasukkan ke dalam form pencarian dan hasil pencarian diperiksa untuk memastikan fitur tersebut berfungsi dengan benar. Fitur lain dapat diuji dengan cara menyesuaikan kode dengan antarmuka dari websitenya.

#### IV. KESIMPULAN

Pendekatan T-Way Testing merupakan metode yang efektif untuk mengoptimalkan pengujian perangkat lunak berbasis web, terutama dalam menghadapi tantangan kombinasi parameter yang kompleks. Teknik ini memungkinkan pengurangan jumlah kasus uji tanpa mengorbankan cakupan pengujian terhadap interaksi antarparameter. Dengan algoritma seperti IPOG, T-Way Testing menghasilkan kombinasi uji yang lebih efisien, sehingga menghemat waktu dan biaya pengujian sambil tetap mempertahankan kualitas dan keandalan perangkat lunak. Keunggulan utama T-Way Testing meliputi efisiensi tinggi, cakupan interaksi yang memadai, penghematan sumber daya, dan fleksibilitas penerapannya pada berbagai skenario pengujian. Namun, pendekatan ini juga memiliki kekurangan, seperti ketergantungan pada pemilihan parameter yang tepat, keterbatasan dalam pengujian skala besar, kebutuhan akan alat bantu khusus, dan kemungkinan pengabaian cacat yang muncul dari kombinasi parameter yang jarang terjadi. Dengan memahami kelebihan dan kekurangan ini, pengembang perangkat lunak dapat memanfaatkan T-Way Testing secara strategis untuk meningkatkan efektivitas pengujian. Ke depan,

inovasi pada algoritma dan alat bantu diharapkan dapat memperluas penerapan T-Way Testing dalam pengujian perangkat lunak yang semakin kompleks dan dinamis.

## V. UCAPAN TERIMA KASIH

Segala puji dan syukur penulis panjatkan ke hadirat Allah SWT atas limpahan rahmat, karunia, dan hidayah-Nya, yang memungkinkan penulis untuk menyelesaikan makalah ini dengan baik. Tanpa rahmat-Nya, penyelesaian makalah ini tentu akan sangat sulit tercapai.

Penulis juga ingin menyampaikan terima kasih yang sebesar-besarnya kepada Bapak Ir. Rila Mandala, M.Eng., Ph.D., selaku dosen pengampu mata kuliah Matematika Diskrit (IF1220), atas ilmu yang telah diberikan selama perkuliahan. Bimbingan, arahan, serta inspirasi yang beliau sampaikan telah memberikan pemahaman yang mendalam tentang konsep-konsep dasar dalam matematika diskrit, khususnya dalam bidang kombinatorika, yang menjadi landasan penting dalam pembuatan makalah ini.

Tak lupa, penulis mengucapkan terima kasih yang tulus kepada Bapak Dr. Ir. Rinaldi Munir, M.T., yang telah mendampingi dengan sabar dan penuh perhatian sejak awal proses pembuatan makalah ini. Bantuan serta masukan yang diberikan sangat berharga bagi penulis dalam mengembangkan materi makalah ini agar dapat tersaji dengan lebih baik dan lebih terstruktur. Semoga segala kebaikan yang telah diberikan mendapat balasan yang setimpal dari Allah SWT.

## REFERENCES

- [1] T. Sh Abdulsamad and M. Y. Potrus, "Achievement of Minimized Combinatorial Test Suite for Configuration-Aware Software Functional Testing Using the Cuckoo Search Algorithm." [Diakses: 4-Jan-2025].
- [2] K. Z. Zamli, F. Din, B. S. Ahmed, and M. Bures, "A hybrid Q-learning sine-cosine-based strategy for addressing the combinatorial test suite minimization problem," *PLoS One*, vol. 13, no. 5, May 2018, doi: 10.1371/journal.pone.0195675. [Diakses: 4-Jan-2025].
- [3] S. Luna Eka Puteri and M. Rafi Raihan Rizal dan Fenni Agustina, "SURVEY TENTANG OPTIMASI KOLONI SEMUT (ANT COLONY OPTIMIZATION) PADA PENGUJIAN PERANGKAT LUNAK," 2020. [Diakses: 5-Jan-2025].
- [4] R. Munir, "Kombinatorika (Bagian 1)," Program Studi Teknik Informatika, Institut Teknologi Bandung, 2024. [Online]. Tersedia: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/18-Kombinatorika-Bagian1-2024.pdf>. [Diakses: 8-Jan-2025].

## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 8 Januari 2025



Andrew Isra Saputra DB (13523110)