

**IMPLEMENTASI METODE *CAUSE EFFECT GRAPHING* PADA  
PENGUJIAN APLIKASI SISTEM INFORMASI ALUMNI FSM  
UNDIP**



**SKRIPSI**

**Disusun Sebagai Salah Satu Syarat  
Untuk Memperoleh Gelar Sarjana Komputer  
Pada Program Studi Ilmu Komputer/Informatika**

**Disusun Oleh :  
Alfi Fadel Majid  
24010314130081**

**PROGRAM STUDI ILMU KOMPUTER/INFORMATIKA  
FAKULTAS SAINS DAN MATEMATIKA  
UNIVERSITAS DIPONEGORO  
2018**

## HALAMAN PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini:

Nama : Alfi Fadel Majid

NIM : 24010314130081

Judul : Implementasi Metode *Cause Effect Graphing* Pada Pengujian Aplikasi Sistem  
Informasi Alumni FSM UNDIP

Dengan ini saya menyatakan bahwa dalam tugas akhir/ skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan di dalam daftar pustaka.

Semarang, 7 Mei 2018

Mahasiswa



Alfi Fadel Majid

NIM. 24010314130081

## HALAMAN PENGESAHAN

Judul : Implementasi Metode *Cause Effect Graphing* Pada Pengujian Aplikasi Sistem  
Informasi Alumni FSM UNDIP

Nama : Alfi Fadel Majid


NIM : 24010314130081

Telah diujikan pada sidang tugas akhir pada tanggal 27 April 2018 dan dinyatakan lulus  
pada tanggal 27 April 2018

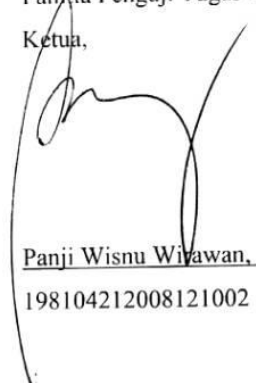
Semarang, 7 Mei 2018

Mengetahui,

Ketua Departemen Ilmu Komputer/ Informatika  
FSM UNDIP

  
Dr. Retno Kusumaningrum, S.Si, M.Kom  
NIP. 198104202005012001

Panitia Penguji Tugas Akhir  
Ketua,

  
Panji Wisnu Wiyawan, ST, MT  
198104212008121002

## HALAMAN PENGESAHAN

Judul : Implementasi Metode *Cause Effect Graphing* Pada Pengujian Aplikasi Sistem  
Informasi Alumni FSM UNDIP

Nama : Alfi Fadel Majid

NIM : 24010314130081

Telah diujikan pada sidang tugas akhir pada tanggal 27 April 2018

Semarang, 7 Mei 2018

Pembimbing



Nurdin Bahtiar, S.Si., MT.

NIP. 19790720 200312 1 002

## ABSTRAK

Pengembangan sistem atau sebuah perangkat lunak melalui beberapa tahapan, diantaranya adalah tahapan pengujian perangkat lunak. Tahapan pengujian bertujuan untuk menjamin mutu dari perangkat lunak dan juga untuk menemukan kesalahan sedini mungkin agar kesalahan tersebut dapat ditangani sesegera mungkin. Untuk menguji suatu sistem atau perangkat lunak, pengembang harus melalui beberapa tahapan dan aspek yang harus dilalui. Salah satu tahapan dan aspek penting dalam pengujian perangkat lunak adalah pembangkitan atau penentuan kasus uji. Kasus uji mengandung sifat-sifat penting dari suatu perangkat lunak yang digunakan untuk mengukur keberhasilan dari perangkat lunak tersebut. Dalam pembangkitan kasus uji terdapat beberapa teknik yang digunakan diantaranya *Whitebox testing* dan *Blackbox testing*. *Blackbox testing* merupakan pengujian yang dapat membangkitkan kasus uji dengan menggunakan spesifikasi kebutuhan perangkat lunak. Pada tugas akhir ini dilakukan pengujian perangkat lunak dengan menggunakan Metode *Cause Effect Graphing* pada Sistem Informasi Alumni FSM UNDIP. Tugas Akhir ini bertujuan untuk mengetahui kebenaran dan hasil reduksi kasus uji yang dihasilkan oleh Metode *Cause Effect Graphing*. Berdasarkan pengujian yang dilakukan pada Sistem Informasi Alumni FSM UNDIP menggunakan metode *Cause Effect Graphing* tidak ditemukannya suatu kesalahan dan kasus uji yang dihasilkan memiliki tingkat kebenaran 100%. Presentase rata – rata kasus uji yang tereduksi pada pengujian Sistem Informasi Alumni FSM menggunakan Metode *Cause Effect Graphing* adalah sebesar 92,04% dari semua kemungkinan kasus uji yang ada.

**Kata Kunci :** Pengujian, Perangkat Lunak, Kasus Uji, *Cause Effect Graphing*

## **ABSTRACT**

*The development of a system or a software is done through several stages, one of them is software testing. The goal of testing phase is to ensure the quality of the software and also to find errors as early as possible so that these errors can be fixed immediately. When testing a system or software, developers have to go through several stages and aspects that must be passed. One of the important stages and aspects of software testing is the generation or determination of test cases. The test case contains the essential properties of a software that are used to measure the success of the software. In the generation of test cases, there are several techniques that can be used, including Whitebox testing and Blackbox testing. Blackbox testing is a test that can generate test cases using software requirements specifications. In this Final Project, software testing is done by using Cause Effect Graphing method on Alumni Information System of FSM UNDIP. The goal of this Final Project is to find out the correctness and the result of test cases reduction produced by the Cause Effect Graphing method. Based on the test conducted on Alumni Information System FSM UNDIP using Cause Effect Graphing method, there is no error found and the test case produced 100% of correctness level. The average percentage of test cases reduced on the testing of Alumni Information System FSM UNDIP using the Cause Effect Graphing method is 92.04% of all possible test cases.*

**Keywords :** *Testing, Software, Test Case, Cause Effect Graphing*

## KATA PENGANTAR

Segala puji syukur penulis panjatkan kepada Allah SWT atas karunia-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “Implementasi Metode *Cause Effect Graphing* Pada Pengujian Aplikasi Sistem Informasi Alumni FSM UNDIP”.

Dalam penyusunan laporan ini tentulah banyak mendapat bimbingan dan bantuan dari berbagai pihak. Untuk itu, pada kesempatan ini penulis mengucapkan rasa hormat dan terima kasih kepada:

1. Ibu Dr. Retno Kusumaningrum, S.Si., M.Kom selaku Ketua Departemen Ilmu Komputer/ Informatika Fakultas Sains dan Matematika Universitas Diponegoro, Semarang.
2. Bapak Helmie Arif Wibawa, S.Si., M.Cs., selaku Koordinator Tugas Akhir Departemen Ilmu Komputer/ Informatika Fakultas Sains dan Matematika Universitas Diponegoro, Semarang.
3. Bapak Nurdin Bahtiar, S.Si., M.T., selaku dosen pembimbing tugas akhir yang telah membantu dalam membimbing dan mengarahkan penulis hingga selesainya skripsi ini.
4. Semua pihak yang telah membantu kelancaran dalam penyusunan tugas akhir, yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa dalam laporan ini masih banyak kekurangan baik dari segi materi ataupun dalam penyajiannya, kritik dan saran sangat penulis harapkan untuk perbaikan pada penulisan ilmiah yang akan datang.

Penulis berharap laporan ini dapat bermanfaat bagi pembaca pada umumnya dan penulis sendiri pada khususnya.

Semarang, 27 April 2018

Penulis,

Alfi Fadel Majid

24010314130081

## DAFTAR ISI

HALAMAN PERNYATAAN KEASLIAN SKRIPSI.....	i
HALAMAN PENGESAHAN .....	ii
ABSTRAK .....	iv
<i>ABSTRACT</i> .....	v
KATA PENGANTAR.....	vi
DAFTAR ISI .....	7
DAFTAR GAMBAR.....	9
DAFTAR TABEL .....	10
DAFTAR ISTILAH.....	12
BAB I PENDAHULUAN .....	13
1.1. Latar Belakang Masalah.....	13
1.2. Rumusan Masalah .....	14
1.3. Tujuan dan Manfaat .....	14
1.4. Ruang Lingkup.....	15
1.5. Sistematika Penulisan.....	15
BAB II LANDASAN TEORI.....	17
2.1. Sistem Informasi Alumni FSM UNDIP.....	17
2.2. Pengujian Perangkat Lunak.....	17
2.3. Metode <i>Cause Effect Graphing</i> .....	19
2.3.1. Pengertian <i>Cause</i> dan <i>Effect</i> .....	19
2.3.2. Pengertian <i>Constraint</i> .....	19
2.3.3. Alur Metode <i>Cause Effect Graphing</i> .....	21
2.4. Model Proses <i>Prototype</i> .....	22
2.5. <i>Automation Testing Selenium</i> .....	23
2.6. Bahasa Pemrograman Java.....	24
BAB III ANALISIS DAN PERANCANGAN SISTEM.....	26
3.1. Deskripsi Umum Perangkat Lunak .....	26
3.1.1. Alur Metode <i>Cause Effect Graphing</i> .....	26
3.2. Kebutuhan Perangkat Lunak .....	36



3.2.1. Model <i>Use case</i> .....	36
3.2.2. <i>Use case</i> skenario .....	38
3.3. Analisis Perangkat Lunak.....	39
3.3.1. <i>Analysis Class</i> .....	40
3.3.2. <i>Sequence Diagram</i> .....	41
3.4. <i>Design</i> .....	43
3.4.1. <i>Class Diagram</i> .....	43
BAB IV IMPLEMENTASI DAN PENGUJIAN .....	44
4.1. Implementasi Sistem .....	44
4.1.1. Spesifikasi Perangkat .....	44
4.1.2. Implementasi <i>Class</i> .....	44
4.2. Pengujian Sistem Informasi.....	45
4.2.1. Skenario Pengujian.....	45
4.2.2. Analisis Hasil Pengujian .....	48
BAB V KESIMPULAN, KETERBATASAN, DAN SARAN .....	55
5.1. Kesimpulan.....	55
5.2. Keterbatasan .....	56
5.3. Saran .....	56
DAFTAR PUSTAKA.....	57
LAMPIRAN - LAMPIRAN .....	58

## DAFTAR GAMBAR

Gambar 2. 1. Model Prototype (Pressman & Maxim, 2001). .....	22
Gambar 3. 1. Tahapan Pada Metode <i>Cause Effect Graphing</i> .....	27
Gambar 3. 2. Grafik Boolean fitur <i>Login</i> .....	30
Gambar 3. 3. <i>Use case</i> Diagram .....	37
Gambar 3. 4. <i>Sequence Diagram</i> Pembangkitan Kasus Uji.....	41
Gambar 3. 5. <i>Sequence Diagram</i> input data <i>Cause</i> .....	42
Gambar 3. 6. <i>Sequence Diagram</i> input data <i>Effect</i> .....	42
Gambar 3. 7. <i>Class Diagram</i> .....	43

## DAFTAR TABEL

Tabel 2. 1. Kondisi <i>Constraint</i> .....	20
Tabel 3. 1. Identifikasi <i>Constraint</i> .....	29
Tabel 3. 2. Jumlah <i>Rule</i> dalam <i>Decision Table</i> .....	31
Tabel 3. 3. Pola Kombinasi <i>Rule</i> dalam <i>Decision Table</i> .....	31
Tabel 3. 4. Penerapan <i>constraint</i> pada <i>cause</i> .....	32
Tabel 3. 5. Reduksi Kolom .....	33
Tabel 3. 6. Mengisi Kolom <i>Effect</i> .....	33
Tabel 3. 7. Mengeliminasi <i>rule</i> yang redundan.....	34
Tabel 3. 8. Hasil Reduksi <i>Rule</i> .....	34
Tabel 3. 9. Pengujian Kasus Uji.....	35
Tabel 3. 10. Daftar <i>Use case</i> .....	36
Tabel 3. 11. <i>Use case</i> Skenario Pembangkitan Kasus Uji .....	38
Tabel 3. 12. <i>Use case</i> Skenario <i>Input Cause</i> .....	38
Tabel 3. 13. <i>Use case</i> Skenario <i>Input Effect</i> .....	39
Tabel 3. 14. Identifikasi <i>Analysis Class</i> .....	40
Tabel 3. 15. <i>Responsibility</i> dan Atribut <i>Class</i> .....	40
Tabel 4. 1. Implementasi <i>Class</i> .....	45
Tabel 4. 2. Fitur Sistem Informasi Alumni FSM UNDIP .....	46
Tabel 4. 3. Spesifikasi Fitur Sistem Informasi Alumni FSM UNDIP .....	46
Tabel 4. 4. <i>Rule</i> fitur <i>login</i> .....	48
Tabel 4. 5. <i>Rule</i> fitur cari data alumni.....	49
Tabel 4. 6. <i>Rule</i> fitur ganti <i>password</i> .....	49
Tabel 4. 7. <i>Rule</i> fitur <i>sign up</i> .....	49
Tabel 4. 8. <i>Rule</i> fitur <i>edit</i> profil.....	50
Tabel 4. 9. Hasil Reduksi Kasus Uji .....	50
Tabel 4. 10. Hasil Eksekusi Kasus Uji Fitur <i>Login</i> .....	51
Tabel 4. 11. Hasil Eksekusi Kasus Uji Fitur Cari Data Alumni .....	51
Tabel 4. 12. Hasil Eksekusi Kasus Uji Fitur Ganti <i>Password</i> .....	52
Tabel 4. 13. Hasil Eksekusi Kasus Uji Fitur <i>Sign up</i> .....	53
Tabel 4. 14. Hasil Eksekusi Kasus Uji Fitur <i>Edit</i> Profil .....	53

Tabel 4. 15. Hasil Eksekusi Kasus Uji dengan Metode <i>Cause Effect Graphing</i> .....	53
Tabel 4. 16. Kebenaran Kasus Uji Setelah Diujikan dengan <i>Selenium</i> .....	54

## DAFTAR ISTILAH

No	Istilah	Deskripsi
1	<i>Constraint</i>	Suatu kondisi yang menjelaskan batasan hubungan antar <i>cause</i> atau antar <i>effect</i>
2	<i>Anchor</i>	Salah satu jenis <i>constraint</i> yang dapat digunakan dalam memberikan batasan pada <i>cause</i> , yang memberikan kondisi nilai 1 ( <i>True</i> ) untuk semua kolom
3	<i>DecisionTable</i>	Tabel kebenaran yang memiliki kolom dengan nilai Boolean, yang digunakan untuk menghasilkan kasus uji
4	<i>Exclusive</i>	Salah satu jenis <i>constraint</i> yang dapat digunakan dalam memberikan batasan pada <i>cause</i> , yang memberikan kondisi dimana dua <i>cause</i> atau lebih tidak bisa sama – sama bernilai 1 ( <i>True</i> )
5	<i>Inclusive</i>	Salah satu jenis <i>constraint</i> yang dapat digunakan dalam memberikan batasan pada <i>cause</i> , yang memberikan kondisi dimana dua <i>cause</i> atau lebih tidak bisa sama – sama bernilai 0 ( <i>False</i> )
6	<i>Mask</i>	Salah satu jenis <i>constraint</i> yang dapat digunakan dalam memberikan batasan pada <i>effect</i> yang memberikan kondisi dimana jika E1 bernilai 1 ( <i>True</i> ) maka E2 menjadi bernilai <i>Mask</i> (tertutupi)
7	<i>One and Only One</i>	Salah satu jenis <i>constraint</i> yang dapat digunakan dalam memberikan batasan pada <i>cause</i> , yang memberikan kondisi dimana dua <i>cause</i> atau lebih tidak bisa sama – sama bernilai 0 ( <i>False</i> ) dan 1 ( <i>True</i> )
8	<i>Require</i>	Salah satu jenis <i>constraint</i> yang dapat digunakan dalam memberikan batasan pada <i>cause</i> , yang memberikan kondisi dimana jika C1 bernilai 1 ( <i>True</i> ) maka C2 menjadi bernilai 1 ( <i>True</i> ), dan sebaliknya
9	<i>Rule</i>	Kolom dari baris ke 1 – n yang terdapat pada <i>decision table</i> yang berisi nilai Boolean

# BAB I

## PENDAHULUAN

Bab ini membahas latar belakang masalah, rumusan masalah, tujuan dan manfaat, dan ruang lingkup dari sistem yang akan dibuat.

### 1.1. Latar Belakang Masalah

Perangkat lunak adalah kumpulan instruksi (program komputer) yang bila dijalankan memberikan fungsi dan kinerja yang diinginkan, struktur data yang memungkinkan program memanipulasi informasi secara memadai, dan dokumen yang menggambarkan operasi dan penggunaan program (Pressman & Maxim, 2001). Untuk mendapatkan suatu perangkat lunak yang berjalan sesuai fungsi yang diinginkan dan mempunyai kualitas yang baik, dibutuhkan beberapa tahapan diantaranya harus melalui proses penjaminan mutu. Di dalam proses penjaminan mutu terdapat suatu aktifitas yaitu pengujian perangkat lunak.

Pengujian perangkat lunak adalah suatu proses pengeksekusian program yang bertujuan untuk menemukan kesalahan (Myers, 2004). Selain itu pengujian perangkat lunak sendiri bertujuan untuk memastikan bahwa program atau perangkat lunak yang sedang diujikan telah berjalan sesuai dengan yang diinginkan. Salah satu aspek penting dalam pengujian adalah pembangkitan kasus uji. Dalam pembangkitan kasus uji terdapat beberapa teknik yang digunakan diantaranya *Whitebox testing* dan *Blackbox testing*.

*Whitebox testing* adalah teknik pengujian yang menggunakan struktur dan perancangan prosedural untuk memperoleh kasus uji (Khan, 2011). Pengujian ini berbasiskan logika atau algoritma dari suatu perangkat lunak dan pengujian ini memungkinkan penguji untuk memeriksa struktur internal perangkat lunak secara langsung. Namun pembangkitan kasus uji dengan teknik ini cenderung sulit diterapkan dan berkemungkinan menimbulkan kesalahan yang berlanjut ketika dilakukan pengujian dengan teknik ini.

Sedangkan *blackbox testing* adalah teknik pengujian tanpa perlu mengetahui struktur internal dari suatu *software* yang akan diuji karena pengujian ini hanya berfokus kepada masukan dan keluaran terhadap spesifikasi suatu *software* (Khan,

2011). Pengujian perangkat lunak ini berbasiskan pada spesifikasi perangkat lunak. Tujuan dari *blackbox testing* itu sendiri adalah untuk menemukan keadaan dimana perangkat lunak tidak berjalan sesuai dengan spesifikasinya tanpa melihat struktur internal dari perangkat lunak tersebut.

Pada teknik *blackbox testing* terdapat beberapa metode yang dapat digunakan untuk membangkitkan kasus uji diantaranya *equivalence partitioning*, *boundary value analysis*, dan *cause effect graphing*. Metode *equivalence partitioning* merupakan pengujian yang membagi data masukan dari perangkat lunak menjadi partisi data yang kemudian diturunkan menjadi kasus uji. Metode *boundary value analysis* merupakan pengujian yang berfokus pada batas nilai – nilai ekstrim yang dipilih. Metode *cause effect graphing* adalah metode pengujian yang membantu dalam membangkitkan kasus uji berdasarkan pada hubungan antar *cause* (masukan) dan *effect* (keluaran) yang terdapat pada spesifikasi kebutuhan perangkat lunak (Mogyorodi, 2003).

Namun, metode *cause effect graphing* dianggap lebih unggul dibandingkan metode yang lainnya karena beberapa hal, diantaranya faktor integrasi antar masukan maupun keluaran yang dapat mereduksi kasus uji sehingga nantinya dapat ditemukan kasus uji (Nursimulu, 1995). Oleh karena itu, pada tugas akhir ini dibangun sebuah aplikasi pembangkit kasus uji Menggunakan Metode *Cause Effect Graphing* untuk menguji spesifikasi dari Sistem Informasi Alumni Mahasiswa Fakultas Sains dan Matematika Universitas Diponegoro.

## **1.2. Rumusan Masalah**

Berdasarkan latar belakang yang telah dijelaskan dapat dirumuskan suatu permasalahan yaitu bagaimana cara dan tahapan dalam pengujian spesifikasi perangkat lunak dengan menggunakan Metode *Cause Effect Graphing* pada Sistem Informasi Alumni Mahasiswa Fakultas Sains dan Matematika Universitas Diponegoro.

## **1.3. Tujuan dan Manfaat**

1. Tujuan dari Tugas Akhir ini adalah :
  - a. Mengetahui apakah Sistem Informasi Alumni Fakultas Sains dan Matematika Universitas Diponegoro sudah berjalan sesuai dengan spesifikasinya dibuktikan dengan menerapkan Metode *Cause Effect Graphing* dalam pengujiannya.

- b. Mengetahui kasus uji yang dihasilkan maupun yang tereduksi dengan Metode *Cause Effect Graphing*.
  - c. Menerapkan ilmu pengetahuan yang telah didapatkan pada aplikasi yang dibuat.
2. Manfaat yang diharapkan dari Tugas Akhir ini adalah :
- a. Bagi Mahasiswa  
Mendapatkan pengetahuan dan pengalaman mengenai pengujian perangkat lunak, terutama dengan Metode *Cause Effect Graphing*.
  - b. Bagi Instansi  
Apabila ditemukan kesalahan pada Sistem Informasi Alumni Fakultas Sains dan Matematika Universitas Diponegoro, maka instansi dapat menjadikannya sebagai bahan evaluasi untuk perbaikan maupun pengembangan sistem kedepannya.

#### **1.4. Ruang Lingkup**

Adapun ruang lingkup masalah di Tugas Akhir ini adalah sebagai berikut :

1. Pengujian dilakukan pada beberapa fitur aplikasi Sistem Informasi Alumni yaitu *login, sign up, cari data alumni, edit profil, dan ganti password*.
2. Pengujian dilakukan menggunakan aplikasi tambahan yaitu *Selenium*.
3. Spesifikasi Sistem Informasi Alumni yang digunakan adalah yang memiliki deskripsi spesifikasi dari modul atau fitur dari Sistem Informasi Alumni.

#### **1.5. Sistematika Penulisan**

Sistematika penulisan yang digunakan dalam penulisan tugas akhir ini terbagi dalam beberapa pokok bahasan, yaitu :

##### **BAB I. PENDAHULUAN**

Bab ini membahas latar belakang masalah, rumusan masalah, tujuan dan manfaat, dan ruang lingkup dari sistem yang akan dibuat.

##### **BAB II. LANDASAN TEORI**

Bab ini menjelaskan teori-teori pengujian yang digunakan oleh penguji dalam membangun dan melaksanakan pengujian perangkat lunak dengan Metode *Cause Effect Graphing* pada Aplikasi Sistem Informasi Alumni FSM Undip.



### BAB III. ANALISIS DAN PERANCANGAN

Bab ini merupakan tahapan dari proses pembangunan perangkat lunak, yaitu tahapan perancangan sistem atau perangkat lunak. Bab ini membahas mengenai deskripsi umum perangkat lunak, analisa dari perangkat lunak hingga perancangan sesuai dengan tahap-tahap pada rekayasa perangkat lunak.

### BAB IV. IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas tentang bagaimana implementasi dari pengembangan Aplikasi Pembangkit Kasus Uji Berbasis Metode *Cause Effect Graphing* dan pengujian yang dilakukan terhadap Sistem Informasi Alumni Fakultas Sains dan Matematika Universitas Diponegoro dengan menggunakan kasus uji yang dihasilkan dari Aplikasi Pembangkit Kasus Uji Berbasis Metode *Cause Effect Graphing*.

### BAB V. PENUTUP

Bab ini membahas tentang kesimpulan dari proses perancangan sampai sistem diuji dan saran untuk pengembangan sistem di masa yang akan datang.