

PEMBANGUNAN ALAT ANALISIS LIPUTAN KOD
DALAM PENGUJIAN PERISIAN

Oleh

MOHD ANNUAR BIN MOHD ISA

Tesis yang diserahkan untuk
memenuhi keperluan bagi
Ijazah Sarjana Sains

APRIL 2011

Penghargaan

Alhamdulillah, syukur ke hadrat Ilahi kerana dengan berkat limpah kurnia-Nya, dapat saya menyiapkan projek penyelidikan ini seterusnya melengkapkan penulisan tesis penyelidikan yang bertajuk ‘Pembangunan Alat Analisis Liputan Kod dalam Pengujian Perisian’. Setinggi-tinggi penghargaan saya ucapkan kepada semua yang terlibat secara langsung mahupun tidak langsung terutamanya kepada pensyarah Universiti Sains Malaysia (USM) yang merangkap penyelia projek saya, Prof. Madya Dr. Kamal Zuhairi Zamli yang tidak mengenal erti jemu dan putus asa dalam membantu saya menyiapkan projek ini. Sepanjang berada di bawah penyeliaan beliau, saya banyak menimba kepakaran serta pengetahuan baru khususnya dalam bidang pengujian perisian ini. Kata-kata perangsang serta nasihat beliau sering kali menjadi titik pendorong serta membekalkan saya dengan semangat baru agar sentiasa bersungguh-sungguh meneruskan penyelidikan saya daripada fasa awal projek hinggalah ke akhir. Pendekatan secara sistematik yang diterapkan oleh beliau dalam menjalankan penyelidikan amat berkesan dan berguna untuk diguna pakai pada masa hadapan. Segala tunjuk ajar serta pertolongan beliau tidak akan saya lupakan hingga akhir hayat, insya-Allah.

Seterusnya saya ingin mengucapkan jutaan terima kasih yang tidak terhingga kepada rakan-rakan seangkatan yang terlibat dalam gagasan penyelidikan perisian di kampus kejuruteraan USM ini termasuklah Muhammad Firdaus Alias, Nor Syarida Jusof, Saidatul Khatimah Said, Mazurawati, Zainal Hisham, Shahrul Afzal, M. Fadel Jamil Klaib, Mohamed I. Younis, F. M Ali dan rakan-rakan lain yang turut sama berusaha untuk memajukan kumpulan ini dengan projek penyelidikan yang berterusan secara

tidak langsung mengharumkan nama USM agar setaraf dengan statusnya sebagai sebuah universiti penyelidikan. Tanpa bantuan mereka, adalah mustahil bagi saya untuk merealisasikan matlamat serta objektif projek penyelidikan ini. Sekalung penghargaan sekali lagi buat mereka.

Tidak dilupakan, kepada ahli keluarga yang tersayang iaitu ayah dan ibu saya yang banyak memohon doa, memberi dorongan serta sokongan moral hingga saya berjaya hingga ke tahap ini. Sesungguhnya jasa mereka berdua tidak akan terbalas dengan pengorbanan saya seumur hidup. Tidak lupa juga abang serta dua orang adik saya yang sentiasa menceriakan hidup saya ketika dalam kemurungan. Saya doakan agar mereka berjaya mencapai impian masing-masing.

Sekalung budi kepada semua warga staf USM kampus kejuruteraan yang tidak jemu-jemu menabur bakti kepada para pelajar agar menjadi insan berguna kepada masyarakat sekaliannya. Projek ini juga tidak mungkin dapat direalisasikan tanpa pembiayaan daripada Kementerian Sains, Teknologi dan Inovasi (MOSTI) melalui dua geran penyelidikan ScienceFund yang bertajuk “Development of a Mobile Agent Based Parallel and Automated JAVA Testing Tool” dan “Development of a Software Fault Injection Tool to Ensure Dependability of COTs for Embedded System Applications. Akhir kata, jutaan terima kasih diucapkan kepada semua yang terlibat menjayakan projek ini sama ada secara langsung ataupun tidak langsung.

Isi Kandungan

Penghargaan	ii
Isi Kandungan	iv
Senarai Jadual.....	viii
Senarai Rajah	x
Terjemahan Istilah	xiii
Singkatan Istilah.....	xvii
Abstrak.....	xviii
Abstract.....	xix
BAB 1	1
PENGENALAN.....	1
1.1 Pengenalan kepada Pengujian	3
1.1.1 Pengujian dalam Kitar Hayat Pembangunan Perisian.....	6
1.1.2 Matlamat Pengujian Perisian.....	8
1.1.3 Analisis Liputan sebagai Mekanisme Pengujian	10
1.2 Motivasi Penyelidikan dan Pernyataan Masalah.....	12
1.3 Matlamat dan Objektif Penyelidikan	17
1.4 Metodologi Penyelidikan	17
1.5 Organisasi Tesis.....	20
BAB 2	22
KAJIAN LITERATUR.....	22
2.1 Pengenalan	22
2.2 Analisis Liputan Kod	24
2.3 Instrumentasi Kod dalam Analisis Liputan.....	29
2.3.1 Fasa Instrumentasi	30
2.3.2 Teknik-teknik Instrumentasi.....	31
2.3.2(a) Instrumentasi Kod Sumber	32
2.3.2(b) Instrumentasi secara Kod Objek	34
2.3.2(c) Instrumentasi secara Dinamik	34
2.3.3 Perbandingan antara Teknik-teknik Instrumentasi	35
2.4 Metrik Liputan Kod	37

2.4.1 Metrik-metrik Liputan Asas	38
2.4.2 Variasi Metrik Asas	44
2.4.3 Kriteria Pemilihan Metrik	50
2.5 Kebergantungan Analisis Liputan terhadap Teknik-teknik Pengujian.....	54
2.5.1 Pengujian Kefungsian	55
2.5.2 Pengujian Struktur	57
2.5.3 Perbandingan antara Pengujian Kefungsian dan Pengujian Struktur	58
2.6 Analisis Liputan Kod dalam Aliran Reka Bentuk Perisian.....	59
2.7 Aliran Proses Analisis Liputan Kod	66
2.8 Peranan Analisis Liputan Kod dalam Pengujian Perisian.....	68
2.9 Alat-alat Analisis Liputan Kod.....	72
2.9.1 Bullseye.....	73
2.9.2 Clover.....	74
2.9.3 EMMA	75
2.9.4 CodeTEST.....	77
2.9.5 gcov.....	78
2.9.6 JCover	79
2.9.7 COVTOOL.....	80
2.9.8 xCover	80
2.10 Penilaian kepada Alat-alat Analisis Liputan Kod.....	81
2.11 Ringkasan Bab.....	84
BAB 3	85
REKA BENTUK SISTEM DAN PELAKSANAAN.....	85
3.1 Pengenalan	85
3.2 Garis Panduan Reka Bentuk Sistem	86
3.3 Modul-Modul Liputan Kod.....	88
3.3.1 Penjana Profil	89
3.3.2 Modul Instrumentasi Kod.....	91
3.3.3 Pengesan Peristiwa.....	93
3.3.4 Pelapor Peristiwa	94
3.4 Pelaksanaan Program Liputan Kod Automatik CovAT.....	94
3.4.1 Mekanisme Penghuraian Kod Sumber Program.....	97

3.4.2 Pengesanan Struktur Aliran Kawalan Kod Program	103
3.4.3 Struktur Penyimpanan Rekod dalam Pangkalan Data	110
3.4.4 Implementasi Instrumentasi Kod Sumber	116
3.4.5 Pengesanan Liputan Kod dalam Pengujian Program.....	128
3.4.6 Pengumpulan Maklumat Liputan dan Penjanaaan Laporan Liputan Kod.....	132
3.4.7 Aplikasi CovAT dalam Domain Pustaka	134
3.5 Ringkasan Bab.....	140
BAB 4	142
KEPUTUSAN DAN PERBINCANGAN.....	142
4.1 Pengenalan	142
4.2 Analisis Liputan dengan CovAT	144
4.3 Pemilihan Kajian Kes	145
4.3.1 Kajian Kes I: Alat Analisis Liputan Kod CovAT.....	147
4.3.1(a) Deskripsi Program.....	147
4.3.1(b) Senarai Kes Ujian.....	148
4.3.1(c) Hasil Pengujian Program menggunakan CovAT	149
4.3.1(d) Pemerhatian dan Perbincangan	152
4.3.2 Kajian Kes II: Program Pengesahan Segi Tiga	154
4.3.2(a) Deskripsi Program.....	155
4.3.2(b) Senarai Kes Ujian.....	157
4.3.2(c) Hasil Pengujian Program menggunakan CovAT	160
4.3.2(d) Pemerhatian dan Perbincangan	161
4.3.3 Kajian Kes II: Program Menu Pesanan Piza Danillo.....	163
4.3.3(a) Deskripsi Program.....	164
4.3.3(b) Eksperimen Pengujian Bahagian 1.....	166
4.3.3(c) Eksperimen Pengujian Bahagian 2.....	177
4.4 Ringkasan Bab.....	183
BAB 5	185
KESIMPULAN	185
5.1 Rumusan.....	185
5.2 Kesimpulan.....	188

5.3 Sumbangan Penyelidikan	191
5.4 Cadangan Penambahbaikan.....	193
5.5 Penutup.....	197
Rujukan	198
Lampiran A : Kod Pseudo.....	205
Lampiran B : <i>Jenny</i>	207
Lampiran C : Senarai Penerbitan.....	211

Senarai Jadual

Jadual 1.1	Definisi-definisi pengujian	3
Jadual 2.1	Perbandingan antara teknik-teknik instrumentasi	36
Jadual 2.2	Perbandingan antara metrik liputan keputusan dan metrik liputan syarat dari segi bilangan kes ujian berdasarkan atur cara contoh 2	43
Jadual 2.3	Hubungan antara metrik liputan modul, fungsi dan blok dari segi liputan kod	53
Jadual 2.4	Rumusan antara pengujian kefungsian dan pengujian struktur	59
Jadual 2.5	Ringkasan ciri-ciri keupayaan alat-alat analisis liputan kod	82
Jadual 3.1	Ringkasan kandungan data profil	119
Jadual 3.2	Perihal kenyataan panggilan yang digunakan dalam proses instrumentasi	124
Jadual 3.3	Lokasi kemasukan kenyataan panggilan instrumentasi bagi dua keadaan unit program	126
Jadual 3.4	Senarai notasi panggilan fungsi individu dalam pustaka CovAT	134
Jadual 4.1	Parameter masukan dan keluaran untuk setiap modul CovAT	148
Jadual 4.2	Senarai kes-kes ujian yang mungkin bagi modul-modul CovAT	149
Jadual 4.3	Rumusan hasil keluaran liputan CovAT untuk analisis ke atas modul-modul utama CovAT	150
Jadual 4.4	Kelas kesetaraan yang mungkin untuk segi tiga yang sah	159
Jadual 4.5	Kelas kesetaraan yang mungkin untuk segi tiga yang tidak sah	159
Jadual 4.6	Rumusan peratusan liputan kod yang dikeluarkan oleh CovAT bagi setiap kes ujian yang dilaksanakan dengan program pengesahan segi tiga	160
Jadual 4.7	Senarai elemen antara muka program menu pesanan piza Danillo	165
Jadual 4.8	Senarai parameter-parameter berserta nilai masukan bagi setiap kategori dalam menu pesanan piza Danillo	167
Jadual 4.9	Kes ujian bagi program piza Danillo untuk eksperimen bahagian 1	170

Jadual 4.10	Peratus liputan yang dikeluarkan oleh CovAT bagi kes ujian janaan <i>Jenny</i> yang dilaksanakan dengan program menu piza Danillo bagi n=1 untuk eksperimen bahagian 1	171
Jadual 4.11	Peratus liputan yang dikeluarkan oleh CovAT bagi kes ujian janaan <i>Jenny</i> yang dilaksanakan dengan program menu piza Danillo bagi n=2 untuk eksperimen bahagian 1	172
Jadual 4.12	Peratus liputan yang dikeluarkan oleh CovAT bagi kes ujian janaan <i>Jenny</i> yang dilaksanakan dengan program menu piza Danillo bagi n=3 untuk eksperimen bahagian 1	173
Jadual 4.13	Senarai parameter-parameter berserta nilai masukan yang baru bagi setiap kategori dalam menu pesanan piza Danillo	178
Jadual 4.14	Kes ujian bagi program menu pesanan piza Danillo bagi tupel n=1 dan n=2 untuk eksperimen bahagian 2	179
Jadual 4.15	Peratus liputan yang dikeluarkan oleh CovAT bagi kes ujian janaan <i>Jenny</i> yang dilaksanakan dengan program menu piza Danillo bagi n=1 untuk eksperimen bahagian 2	180
Jadual 4.16	Peratus liputan yang dikeluarkan oleh CovAT bagi kes ujian janaan <i>Jenny</i> yang dilaksanakan dengan program menu piza Danillo bagi n=2 untuk eksperimen bahagian 2	181

Senarai Rajah

Rajah 1.1	Kitar hayat pembagunan produk perisian	6
Rajah 1.2	Aliran keseluruhan metodologi penyelidikan	19
Rajah 2.1	Petikan atur cara contoh 1	40
Rajah 2.2	Petikan atur cara contoh 2	41
Rajah 2.3	Petikan atur cara contoh 3	44
Rajah 2.4	Petikan atur cara contoh 4	46
Rajah 2.5	Aliran kawalan sistem perisian bermodul	47
Rajah 2.6	Petikan atur cara contoh 5	49
Rajah 2.7	Aliran reka bentuk sistem perisian	60
Rajah 2.8	Analisis liputan secara berperingkat	62
Rajah 2.9	Peratusan penentusahan reka bentuk perisian melawan masa	64
Rajah 2.10	Peratusan penentusahan reka bentuk perisian melawan masa melalui analisis liputan kod	65
Rajah 2.11	Carta aliran umum analisis liputan kod	66
Rajah 2.12	Paparan menu antaramuka MS Excel 2003	69
Rajah 3.1	Modul-modul CovAT	88
Rajah 3.2	Modul Penjana Profil	90
Rajah 3.3	Sebahagian atur cara penghurai kod	100
Rajah 3.4	Masukan dan keluaran modul penjana profil	103
Rajah 3.5	Deklarasi fungsi dalam pengaturcaraan C dan C++	104
Rajah 3.6	Gambar rajah aktiviti bagi proses penghuraian kod sumber	110
Rajah 3.7	Aliran akses pangkalan data oleh modul-modul utama CovAT	111
Rajah 3.8	Struktur asas perwakilan data dalam fail data janaan modul CovAT	112
Rajah 3.9	Fail data prestasi yang dijana melalui pelaksanaan kod sumber instrumentasi	117
Rajah 3.10	Aliran proses instrumentasi CovAT	118

Rajah 3.11	Keluaran profil program oleh modul penjana profil	119
Rajah 3.12	Pemecahan struktur data profil program kepada struktur asas	121
Rajah 3.13	Struktur data profil yang mudah difahami pengguna	122
Rajah 3.14	Kod pseudo rutin pengesanan unit yang mempunyai fungsi 'main'	126
Rajah 3.15	Contoh sebahagian kenyataan panggilan instrumentasi yang dimasukkan ke dalam kod sumber secara terus	127
Rajah 3.16	Gambar rajah aktiviti bagi proses instrumentasi kod sumber	128
Rajah 3.17	Maklumat suap balik masukan bagi setiap pelaksanaan kod sumber program dengan kes ujian	129
Rajah 3.18	Masukan dan keluaran modul pengesanan peristiwa	130
Rajah 3.19	Gambar rajah aktiviti bagi proses pengesanan prestasi liputan kod	132
Rajah 3.20	Gambar rajah aktiviti bagi proses penjanaan laporan liputan kod	133
Rajah 3.21	Contoh tettingkap antara muka CovAT yang memaparkan rumusan prestasi pelaksanaan program	133
Rajah 3.22	Operasi penghuraian struktur kod sumber melalui anotasi fungsi penjana profil dari pustaka CovAT	137
Rajah 3.23	Operasi instrumentasi kod melalui anotasi fungsi instrumentasi dari pustaka CovAT	138
Rajah 3.24	Operasi pengesanan liputan kod melalui anotasi fungsi pengesanan peristiwa instrumentasi dari pustaka CovAT	139
Rajah 4.1	Carta aliran proses liputan kod CovAT	144
Rajah 4.2	Graf peratus liputan melawan bilangan kes ujian bagi keluaran liputan bagi setiap modul CovAT	151
Rajah 4.3	Aliran operasi program pengesanan segi tiga	157
Rajah 4.4	Program pengesanan segi tiga	158
Rajah 4.5	Graf peratus liputan melawan bilangan kes ujian bagi keluaran liputan program pengesanan segi tiga oleh CovAT	161
Rajah 4.6	Menu antara muka program menu pesanan piza Danillo	165
Rajah 4.7	Contoh janaan <i>Jenny</i> bagi 12 parameter, 2 nilai untuk 12 tupel	169

Rajah 4.8	Contoh janaan <i>Jenny</i> bagi 12 parameter, 2 nilai untuk satu tupel	169
Rajah 4.9	Graf peratus liputan melawan bilangan kes ujian bagi keluaran liputan program menu pesanan Danillo untuk janaan kes ujian <i>Jenny</i> $n=1$ untuk eksperimen bahagian 1	172
Rajah 4.10	Graf peratus liputan melawan nombor kes ujian bagi keluaran liputan program menu pesanan Danillo untuk janaan kes ujian <i>Jenny</i> $n=2$ untuk eksperimen bahagian 1	173
Rajah 4.11	Graf peratus liputan melawan bilangan kes ujian bagi keluaran liputan program menu pesanan Danillo untuk janaan kes ujian <i>Jenny</i> $n=3$ untuk eksperimen bahagian 1	174
Rajah 4.12	Petikan laporan janaan CovAT yang menunjukkan rumusan prestasi pelaksanaan kod program dan elemen kod sumber yang belum dijalankan oleh kes ujian	176
Rajah 4.13	Graf peratus liputan melawan bilangan kes ujian bagi keluaran liputan program menu pesanan Danillo untuk janaan kes ujian <i>Jenny</i> $n=1$ untuk eksperimen bahagian 2	181
Rajah 4.14	Graf peratus liputan melawan bilangan kes ujian bagi keluaran liputan program menu pesanan Danillo untuk janaan kes ujian <i>Jenny</i> $n=2$ untuk eksperimen bahagian 2	182

Terjemahan Istilah

Bahasa Melayu

Aksara
Aliran kawalan
Analisis impak pindaan
Analisis nilai sempadan
Baris arahan
Binaan semantik
Blok
Boleh Laku
Cabang
Diagnosis ralat
Elisitasi
Fail pengepala
Fail sambungan
Fungsi
Garis perintah
Gelung tak terhingga
Juruwang berautomat
Kajian kes
Kata kunci
Kebergantungan
Kebolehgunaan semula
Kebolehharapan
Kebolehlanjutan
Kebolehsenggaraan
Kecekapan
Kekangan
Kekompleksan
Kelas kesetaraan
Kemudahan
Kenyataan kembali

Bahasa Inggeris

Character
Control-flow
Change impact analysis
Boundary value analysis
Command line
Semantic construct
Block
Executable
Branch
Error diagnostics
Elicitation
Header file
Extension
Function
Command line
Infinite loop
Automated teller machine
Case study
Keyword
Dependency
Reusability
Dependability
Extensibility
Maintainability
Efficiency
Limitation/constraint
Complexity
Equivalence class
Portability
Return statement

Kenyataan lemparan	Throw statement
Kenyataan majmuk	Compound statement
Kenyataan panggilan	Call statement
Kenyataan panggilan instrumentasi	Instrumentation call
Kenyataan tukaran	Switch statement
Kes Ujian	Test case
Ketaknormalan	Abnormality
Keutuhan	Reliability
Kod bait	Byte code
Kod pseudo	Pseudo code
Kod sumber	Source code
Koma bernoktah	Semicolon
Kompil	Compile
Laku / Laksana	Run
Lalai	Default
Lewahan	Redundancy
Liputan baris	Line coverage
Liputan kenyataan	Statement coverage
Liputan keputusan/cabang	Decision/ branch coverage
Liputan Kod	Code coverage
Liputan laluan	Path coverage
Liputan syarat	Condition coverage
Logik-ATAU	Logical-OR
Logik-DAN	Logical-AND
Masa jalanan	Run time
Menganotasi	Annotate
Mengasal	Initialize
Mesin maya	Virtual machine
Nyahpeijat	Debug
Pangkalan data	Database
Pemalar	Constant
Pemautan	Linking
Pembilang	Counter

Pemboleh ubah	Variable
Pemetakan kesetaraan	Equivalence partitioning
Peminimuman	Minimization
Pemprosesan peristiwa	Event processing
Penatalan mengufuk	Horizontal scrolling
Pendakap	Brace
Penentusahan	Verification
Pengaturcaraan berorientasi objek	Object-oriented programming
Pengaturcaraan berstruktur	Structured programming
Pengawas	Monitor
Pengelola sampukan	Interrupt handler
Pengesahsahihan	Validation
Pengesan peristiwa	Event tracer
Penghurai kod	Code parser
Pengkompil	Compiler
Pengoptimuman	Optimization
Pengujian kefungsiian	Functional testing
Pengujian mutasi	Mutation testing
Pengujian pembaikan	Improvement testing
Pengujian rawak	Random testing
Pengujian regresi	Regression testing
Pengujian struktur	Structural testing
Pengujian unit	Unit testing
Pengutamaan	Prioritization
Penjana profil	Profiler
Penunding	Pointer
Penunding nol	Null pointer
Peranti ingatan	Memory Device
Perilaku logik	Logical behavior
Perkakasan	Hardware
Peruntukan memori	Memory allocation
Pincang Tugas	Malfunction
Pindaan kod	Code instrumentation

Prapemproses	Preprocessor
Prom perintah	Command prompt
Pustaka	Library
Rentetan	String
Rutin	Routine
Salinan sandar	Backup copy
Seni bina terbuka	Open architecture
Sistem sendiri	Standalone system
Sistem terbenam	Embedded system
Sokongan	Backup
Strim	Stream
Suap balik	Feedback
Surih	Trace
Suruhan tokokan	Increment instruction
Syarat	Condition
Tanda kurung kurawal	Curly bracket
Tatar	Upgrade
Tatasusunan	Array
Tetingkap paparan	Display window
Tuntas	Exhaustive
Ulasan	Comment
Unsur	Element

Singkatan Istilah

<i>Singkatan</i>	<i>Maksud</i>
ANSI	American National Standards Institute
API	Application Programming Interface
ATM	Automated Teller Machine
EOF	End of File
GCC	GNU Compiler Collection
GUI	Graphical User Interface
JDK	Java Development Kit
JRE	Java Runtime Environment
LOC	Physical Line of Code
MS Excel	Microsoft Office Excel
OS	Operating System
SMS	Short Message Services
USD	United States Dollar
V & V	Verification and Validation

PEMBANGUNAN ALAT ANALISIS LIPUTAN KOD
DALAM PENGUJIAN PERISIAN

Abstrak

Walaupun diingini, ujian perisian tuntas secara praktikal adalah tidak mungkin kerana kekangan sumber dan masa. Seringkali, analisis liputan dengan sokongan alat yang sesuai diterima pakai sebagai salah satu penunjuk kesempurnaan ujian (iaitu sama ada untuk meneruskan lebih lanjut atau tidak ataupun untuk menghentikan sepenuhnya pengujian). Selama bertahun-tahun, banyak alat-alat analisis liputan telah dibangunkan dengan kejayaan yang berbeza-beza. Untuk memperbaiki dan meningkatkan kebolegunaan sedia ada, kajian ini menganalisis alat-alat yang ada dalam usaha untuk mengenal pasti teknik yang sesuai dan metrik liputan untuk pelaksanaan serta ciri-ciri untuk dipertimbangkan. Analisis ini telah menghasilkan reka bentuk dan implementasi CovAT. CovAT dibangunkan bercirikan reka bentuk modular dengan menggunakan instrumentasi kod sumber. Dengan menggunakan CovAT, liputan kod boleh dianalisis secara automatik menggunakan metrik unit, fungsi dan blok yang mewakili tahap hierarki liputan. Dengan memberi penekanan kepada kriteria kebolehlanjutan dan kebolegunaan semula kod, seni bina sistem terbuka CovAT juga menawarkan sokongan operasi dalam domain pustaka selain daripada kemampuan sendiri. Keputusan daripada tiga kajian kes membuktikan bahawa CovAT mempunyai ciri-ciri mencukupi untuk menjalankan analisis liputan kod dengan sempurna berdasarkan kriteria kecukupan pengujian yang disasarkan.

DEVELOPMENT OF CODE COVERAGE ANALYSIS TOOL IN SOFTWARE TESTING

Abstract

Although desirable, exhaustive software testing is practically impossible due to resource and timing constraints. Often, coverage analysis with the support of a suitable tool is adopted as one of the test completeness indicators (i.e. whether or not to continue further or to completely stop testing). Over the years, many useful coverage analysis tools have been developed with varying success. In order to improve and enhance their applicability, this research analyzes the existing tools in an effort to identify the suitable technique and coverage metrics for implementation as well as features for consideration. This analysis has resulted in to the design and implementation of CovAT. CovAT is developed based on modular design using source code instrumentation. Using CovAT, code coverage can be automatically analyzed using unit, function and blok metrics that represent hierarchical level of coverage. Emphasizing on extensibility and code reusability criteria, the open-architecture design of CovAT also offers operational support in library domain apart from its standalone capabilities. Results of three case study evaluations prove that CovAT has sufficiently enough feature to conduct complete code coverage analysis based on the targeted test-adequacy criteria.

BAB 1

PENGENALAN

Seiring dengan kecanggihan teknologi pada masa kini, evolusi dalam sektor pembangunan perisian juga turut berkembang dengan pesat. Hari demi hari, kebanyakan aplikasi harian yang dahulunya bergantung kepada perkakasan, kini telah digantikan dengan aplikasi perisian yang terbukti lebih cekap serta mudah digunakan. Daripada transformasi telegraf yang digunakan pada era 1830-an untuk menghantar mesej bertulis kepada aplikasi servis pesanan ringkas atau SMS dalam telefon mudah alih kini, perisian telah menjadi tunjang utama dalam proses permodenan dunia. Peralihan permintaan pengguna kepada aplikasi perisian ini secara meluas dan pesat dalam masa yang singkat adalah sememangnya dijangka atas dasar ia lebih menguntungkan untuk jangka masa panjang kerana sistem perisian adalah lebih mudah untuk dipinda serta dikemaskini mengikut keperluan semasa pengguna. Keboleharapan sesebuah perisian juga adalah tinggi jika dibandingkan dengan perkakasan oleh sebab ia tidak mudah rosak ataupun pincang tugas akibat digunakan dengan terlalu lama ataupun ketika beroperasi dalam persekitaran yang tidak sesuai. Lantaran yang demikian, aplikasi perisian berupaya menjimatkan kos serta tenaga kerja berbanding operasi perkakasan yang memerlukan penyelenggaraan secara berulang kali dan menyeluruh bagi memastikan operasi pengendaliannya pada tahap yang optimum.

Walaupun begitu, ciri-ciri perisian yang mudah dipinda mahupun dikemaskini tersebut sering menimbulkan kebimbangan di kalangan penyelidik dari segi keboleharapan dan keutuhan perisian terbabit. Lantaran itu, bidang pengujian

perisian adalah satu elemen utama dalam menjalankan pengesahsahihan sesebuah program atau sistem perisian. Pengujian yang merangkumi majoriti fasa pembangunan perisian dan meliputi 40 hingga 50 peratus daripada kos pembangunan keseluruhan (Beizer, 1990; Damm, Olsson, & Lundberg, 2005) ini adalah amat penting lebih-lebih lagi dalam aplikasi sistem perisian kritikal yang melibatkan keselamatan nyawa pengguna seperti perisian pengimejan komputer bagi tujuan perubahan, sistem beg udara dalam kenderaan dan sebagainya.

Pengujian yang tidak mencukupi mengundang bahaya kepada pengguna akhir, pekerja pemasangan dan penyelenggaraan perisian serta kerugian dari segi kewangan disebabkan oleh pembaziran tenaga kerja dan kelewatan bagi membolehkan sistem berkenaan beroperasi (Gitzel, Krug, & Brhel, 2010; Naur & Randell, 1969). Sebagai contoh, pada tahun 2005, sebuah syarikat peruncit makanan terkenal British, J Sainsbury PLC mengalami kerugian sebanyak USD 526 juta dalam pelaburan ke atas sebuah sistem pengurusan rantai bekalan automatik apabila sistem tersebut pincang tugas ketika operasi (Charette, 2005). Akibat kegagalan sistem berkenaan, pengagihan barang dagangan yang diurus oleh sistem berkenaan terhenti di gudang milik syarikat terbabit dan tidak dapat diagihkan ke kedai-kedai sasaran. Lantaran itu, syarikat terbabit terpaksa pula mengupah 3000 pekerja tambahan bagi mengagihkan stok barangan yang banyak tersebut secara manual. Kekurangan pengujian terhadap sistem telah dikenal pasti sebagai punca utama kegagalan sistem tersebut. Maka adalah mandatori bagi pihak berkepentingan dalam pembinaan perisian ini bagi mencapai tahap pengujian yang mencukupi sebelum produk mereka sampai ke tangan pengguna agar kualiti serta ciri-ciri keselamatannya betul-betul terjamin.

1.1 Pengenalan kepada Pengujian

Sebelum mendalami proses pengujian secara lebih terperinci, adalah lebih baik sekiranya kita memahami terlebih dahulu definisi pengujian yang diberikan oleh beberapa penyelidik terkemuka dalam bidang perisian daripada era masing-masing.

Jadual 1.1 menyenaraikan definisi pengujian daripada tiga era yang berbeza.

Jadual 1.1 Definisi-definisi pengujian

Definisi
“Testing is the process of executing a program with the intent of finding errors” (Myers, 2004)
“Testing is any activity aimed at evaluating an attribute of a program or system. Testing is the measurement of software quality” (Hetzel, 1998)
“Testing is a concurrent lifecycle process of engineering, using, and maintaining testware in order to measure and improve the quality of the software being tested” (Craig & Jaskiel, 2002)

Merujuk kepada Jadual 1.1, Myers berpendapat bahawa pengujian merupakan proses melaksanakan sesuatu program dengan tujuan mencari ralat di dalamnya (Myers, 2004). Pada ketika bukunya ditulis iaitu pada tahun 1979, definisinya itu adalah antara olahan yang terbaik dan paling hampir mencerminkan proses pengujian ketika itu. Dalam huraian beliau yang lebih lanjut, pengujian merupakan proses yang berlaku di penghujung kitar pembangunan perisian dan tujuannya adalah untuk mencari ralat yang hadir di dalamnya.

Hetzel pula berpendapat bahawa pengujian adalah mana-mana aktiviti yang memfokuskan penilaian atribut sesuatu program atau sistem dan ia merupakan pengukuran terhadap kualiti perisian (Hetzel, 1998). Berbanding era sebelumnya, pada tahun 1983, definisi mengenai proses pengujian telah diubah suai untuk mengambil kira penilaian terhadap kualiti sesebuah perisian selain proses mengesan kecacatan atau ralat.

Dalam pada itu, dalam beberapa tahun lepas iaitu pada tahun 2002, Craig dan Jaskiel mendapati terdapat sedikit kekurangan dalam dua definisi sebelum ini atas alasan definisi tersebut telah mengecilkan ruang lingkup pengujian perisian (Craig & Jaskiel, 2002). Mereka beranggapan bahawa definisi-definisi tersebut masih boleh diguna pakai hingga ke hari ini hanya atas dasar terdapat aspek-aspek yang melibatkan pengujian perisian. Bagi menutup kekurangan terbabit, mereka mencapai kata sepakat untuk memberi definisi pengujian sebagai proses jangka hayat serempak yang melibatkan kejuruteraan, menggunakan serta menyelenggara perkakas ujian bagi mengukur dan memperbaiki kualiti perisian yang diuji. Melalui definisi ini, tidak dinyatakan secara terus tentang pengesanan ralat walaupun ia masih matlamat pengujian yang sah. Selain itu, definisi mereka juga turut mengambil kira bukan sahaja aspek mengukur tahap kualiti sesebuah perisian malah turut melibatkan penambahbaikan terhadap kualiti perisian terbabit.

Walau bagaimanapun, definisi pengujian yang lebih umum serta dijadikan rujukan secara meluas adalah berpandukan kepada jilid glosari piawai terminologi kejuruteraan perisian yang dikeluarkan oleh *Institute of Electrical and Electronic Engineers* (IEEE) yang bertujuan merekodkan serta mengenal pasti terma-terma

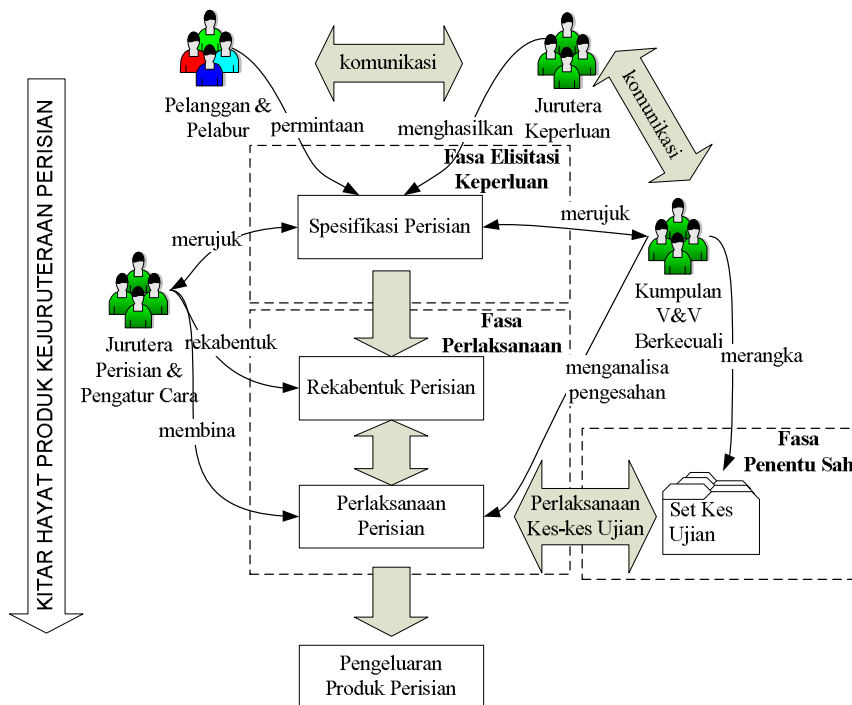
yang diguna pakai dalam bidang kejuruteraan perisian pada masa kini. Mengikut piawaian tersebut, pengujian perisian diterjemahkan sebagai proses menganalisis perisian untuk mengesan perbezaan (iaitu kecacatan atau pepijat) antara spesifikasi yang diperoleh dengan spesifikasi yang ditetapkan pada awalnya serta bagi tujuan menilai ciri-ciri perisian.

Pada dasarnya, definisi ini masih memperakui bahawa pengesanan ralat sebagai salah satu motif pengujian. Tujuan lainnya adalah untuk menilai ciri-ciri yang hadir pada perisian secara tidak langsung merupakan satu mekanisme bagi mengukur tahap prestasi dan kualiti sesebuah perisian. Definisi ini secara menyeluruh, menyokong idea bahawa pengujian adalah satu proses mengendalikan perisian agar memenuhi syarat-syarat tertentu yakni pertamanya adalah untuk mengesahkan bahawa perisian beroperasi seperti yang dikehendaki, keduanya adalah bagi mengesan ralat, dan yang terakhirnya adalah untuk mengesahkan sama ada spesifikasi yang dicapai adalah menurut kemahuan pengguna ataupun tidak.

Kesimpulannya, pengujian adalah satu mekanisme pengukuran kualiti sistem perisian dengan cara menjalankan proses program dengan tujuan untuk mengesan kehadiran pepijat yang dapat menghindarkan aliran operasi program. Ia juga dapat menilai sejauh mana reka bentuk program telah menyimpang daripada spesifikasi asal selain mengurangkan risiko kegagalan sistem perisian pada akhir fasa pembangunan.

1.1.1 Pengujian dalam Kitar Hayat Pembangunan Perisian

Sebelum penerangan yang lebih lanjut mengenai proses pengujian perisian ini, adalah lebih baik jika terlebih dahulu kita meninjau bagaimana proses pengujian itu dapat dimanfaatkan dalam keseluruhan kitar hayat pembangunan perisian. Merujuk kepada Rajah 1.1, kitar hayat pembangunan produk perisian melibatkan proses penghasilan perisian yang meliputi semua peringkat termasuklah hal ehwal pengurusan, spesifikasi permintaan pengguna, reka bentuk, proses pengujian, sistem sokongan perisian, penyelenggaraan produk dan sebagainya.



Rajah 1.1 Kitar hayat pembagunan produk perisian (Zamli et al., 2009)

Ia juga turut melibatkan interaksi antara golongan berkepentingan berbeza daripada pelbagai peringkat pengurusan mahupun teknikal dalam struktur organisasi projek termasuklah para pelabur, para pelanggan dan juga pembekal yang bertindak sebagai

tunjang utama yang menyokong sesebuah projek pembangunan perisian (Sudarsan, Fenves, Sriram, & Wang, 2005). Komunikasi yang baik dan telus dalam struktur organisasi ini adalah penting bagi menjamin operasi pembangunan yang lancar dan mengikuti prosedur yang ditetapkan serta dapat dilaksanakan mengikut ketetapan jangka masa seperti yang dirancang.

Secara khususnya, pembangunan perisian bermula dengan fasa elisitasi keperluan yang melibatkan usaha sama antara pengguna dan para pelabur dengan jurutera keperluan untuk menghasilkan spesifikasi sistem perisian yang hendak dicapai dalam produk akhir berdasarkan permintaan yang telah dipersetujui antara pihak berkenaan. Kemudiannya, dengan bersandarkan kepada spesifikasi yang telah dibuat, jurutera perisian dan pengatur cara akan mengembleng tenaga untuk menghasilkan reka bentuk perisian terbabit untuk diimplementasikan dalam fasa berikutnya iaitu fasa pelaksanaan.

Pada dasarnya, proses pengujian dipraktikkan dalam fasa pengesahsahihan yang berlaku seiring dengan kedua-dua fasa sebelum ini. Kumpulan berkecuali yang bertanggungjawab untuk melakukan pengujian ini digelar sebagai kumpulan penentusahan dan pengesahsahihan. Jurutera V&V ini akan merujuk terlebih dahulu jurutera keperluan bagi mengetahui spesifikasi asal perisian. Seterusnya, mereka akan membina set kes ujian berdasarkan spesifikasi tersebut untuk diguna pakai sewaktu pelaksanaan sistem perisian yang telah siap diimplimentasikan. Melalui pengujian sistem perisian dengan set kes ujian yang dibina berdasarkan spesifikasi asal, maka mereka dapat mengetahui sejauh mana reka bentuk produk akhir perisian menyimpang daripada spesifikasi yang dimahukan oleh pengguna. Sekiranya

pelaksanaan dengan set kes ujian memenuhi spesifikasi keperluan, maka produk perisian adalah menepati kehendak pengguna dan sedia untuk dipasarkan. Jika perkara sebaliknya berlaku, maka reka bentuk dan proses implementasi perlu disemak semula hinggalah pengujian dengan set ujian mengeluarkan hasil yang positif.

Selain itu, pengujian dalam fasa pembangunan perisian membolehkan golongan atasan yang berkepentingan seperti pelabur mahupun para pelanggan untuk sentiasa peka dan faham akan perjalanan proses reka bentuk perisian sama ada mengikut kemahuan mereka ataupun sebaliknya. Dengan cara ini, mereka boleh menjangkakan lebih awal hasil akhir pembangunan perisian terbabit dan berkesempatan untuk membetulkan kelemahan dari segi amalan pengaturcaraan secara keseluruhannya ataupun untuk mengadakan perbincangan semula mengenai spesifikasi perisian yang kurang memuaskan bagi mereka (Kaner, Glass, Collard, Bertolino, & Bach, 2006) secara tidak langsung meningkatkan hasil kualiti akhir produk perisian yang dibangunkan.

1.1.2 Matlamat Pengujian Perisian

Dalam sesebuah projek pembangunan perisian, memang tidak boleh dinafikan peranan proses pengujian dalam mengenal pasti permasalahan yang hadir ketika pelaksanaan sama ada dari segi luaran mahupun struktur dalaman sistem perisian yang memfokuskan kepada proses pengesanan pepijat sewaktu pengujian dijalankan (Luo, 2001). Pengujian sepatutnya dapat memberikan diagnosis yang jelas agar sebarang kecacatan atau pepijat dapat dibetulkan sebelum sesebuah perisian

dibenarkan untuk berganjak ke fasa pembangunan seterusnya. Secara tidak langsung, pengujian berupaya meminimakan tahap risiko kegagalan perisian pada peringkat yang lebih lanjut sekiranya kebanyakan pebijat dapat dikesan lebih awal selain memastikan sesebuah program itu berfungsi atau tidak. Pengujian juga merupakan satu medium bagi jurutera perisian untuk mengenal pasti sejauh mana reka bentuk perisian mereka menyimpang daripada spesifikasi asal agar program yang dibangunkan boleh menjalankan operasi yang sepatutnya.

Namun begitu, bagi mendokong matlamat seperti yang dinyatakan, beberapa kriteria asas pengujian hendaklah dilaksanakan terlebih dahulu. Secara ringkasnya, pengujian merupakan satu proses pengendalian perisian yang bersandarkan kepada kriteria-kriteria asas berikut:

- i. Proses Penentusahan

Pemeriksaan dijalankan ke atas perisian bagi mengesahkan bahawa perisian yang sedang dibina pada fasa-fasa tertentu adalah konsisten dan menepati spesifikasi keperluan yang dinyatakan pada awalan fasa pembangunan projek tersebut (Naik & Tripathy, 2008). Proses pengesahan ini tidak semestinya hanya dilakukan ketika fasa akhir pembinaan produk perisian malahan turut merangkumi pengesahan fasa pertengahan seperti fasa spesifikasi keperluan, spesifikasi reka bentuk, penulisan kod mahupun dokumentasi manual pengguna. Dalam kata lain, proses ini mengesahkan sama ada perisian yang dibina adalah berdasarkan spesifikasi yang dinyatakan pengguna.

ii. Proses Pengesanan Ralat

Pengujian terhadap sesebuah perisian berkemungkinan melibatkan kemasukan input-input tidak lazim yang membuatkan program pincang tugas. Hal yang demikian bukanlah untuk merencatkan perjalanan perisian terbabit, namun bertindak sebagai satu mekanisme pengesanan pepijat yang hanya dapat dijejak melalui pelaksanaannya dengan input-input luar kawalan terbabit agar pepijat ini kemudiannya boleh ditangani dengan lebih efisien. Sebagai contoh, suatu sistem yang hanya menggunakan integer sebagai input akan pincang tugas apabila pengguna memasukkan aksara huruf atau abjad sebagai input. Dalam konteks ini, tindakan pengguna terbabit boleh dikira sebagai proses pengesanan ralat kerana berupaya untuk mendedahkan kelemahan atau pepijat dalam sistem terbabit.

iii. Pengesahsahihan

Proses ini merujuk kepada aktiviti mengesahkan ketepatan dalam sistem perisian atau secara ringkasnya proses menentukan sama ada sistem perisian berfungsi sebagaimana yang dikehendaki oleh pengguna mahupun tidak (Adrion, Branstad, & Cherniavsky, 1982). Dibandingkan dengan proses pengesanan yang dinyatakan dalam (a), aktiviti ini lebih memfokus kepada produk akhir yang dijalankan secara terperinci dari sudut pandangan pengguna.

1.1.3 Analisis Liputan sebagai Mekanisme Pengujian

Seperti yang dihuraikan dengan panjang lebar dalam subtopik sebelum ini, pengujian perisian adalah lebih daripada sekadar proses untuk mengenal pasti pepijat. Ia juga sebenarnya adalah sebahagian daripada mekanisme penentu jaminan kualiti perisian.

Secara idealnya, perisian yang berkualiti boleh diterjemahkan sebagai perisian yang kurang pepijat, mematuhi spesifikasi keperluan atau jangkaan pengguna, mudah diselenggarakan, dihantar kepada pengguna tepat pada masa dan berada dalam lingkungan kos yang diperuntukkan. Namun begitu, ini tidak bererti pengujian itu sendiri dapat memastikan sesebuah perisian itu memenuhi tahap kualiti tersebut, kerana pada dasarnya kualiti merupakan satu kriteria yang subjektif. Ia mempunyai kebergantungan terhadap faktor golongan pengguna dan pengaruhnya terhadap keseluruhan perancangan projek pembangunan perisian. Pengguna daripada satu kumpulan tertentu mempunyai perspektif dan pandangan berbeza terhadap kualiti sesebuah perisian jika dibanding dengan kumpulan yang lain (Zamli et al., 2009).

Jika dilihat daripada aspek yang lebih luas, sesebuah projek pembangunan perisian selalunya melibatkan kumpulan pengguna daripada pelbagai peringkat yang berbeza sebagai contoh pengurus projek pembangunan, penguji perisian, eksekutif pemasaran, jurutera penyelenggara perisian dan juga pengguna akhir. Mengikut kelazimannya, pengguna akhir lebih mementingkan perisian yang mesra pengguna serta bebas daripada pepijat manakala para pengurus projek pula menilai kualiti produk mereka dari segi keuntungan hasil penjualan perisian terbabit dan begitulah seterusnya.

Lantaran itu adalah sukar dan mustahil untuk menjalankan proses pengujian bagi mencapai tahap kualiti yang dapat memenuhi kehendak kesemua individu terbabit dalam sesuatu projek pembangunan yang dikekang oleh jangka masa tertentu serta matlamat pengujian yang berbeza untuk setiap fasa pembangunan perisian. Tanpa satu mekanisme pengujian yang dapat menggambarkan kualiti sesebuah perisian yang

dibangunkan, kesemua golongan tersebut tidak mampu menentukan tahap prestasi pengujian program mereka pada fasa-fasa tertentu dalam pembangunan. Walau bagaimanapun, dengan kemajuan bidang penyelidikan perisian yang semakin berkembang kini, beberapa mekanisme telah dikenal pasti untuk menyelesaikan permasalahan tersebut. Salah satu mekanisme yang sering digunakan bagi menggambarkan prestasi pengujian perisian adalah melalui analisis liputan kod.

Analisis liputan kod merupakan kaedah terbaik untuk menggambarkan secara jelas kualiti sesebuah perisian dari segi kuantitatif melalui pengukuran tahap prestasi pelaksanaan struktur aliran kawalan kod program sewaktu pengujian. (Lawrance, Clarke, Burnett, & Rothermel, 2005; Lingampally, Gupta, & Jalote, 2007). Selain dapat membantu pengatur cara dalam proses penyahpejatan melalui pengesanan kehadiran pepijat dalam kod sumber program dengan lebih awal, analisis liputan kod ini juga merupakan elemen pengujian yang penting dalam menyokong proses penentusahan serta pengesahsahihan yang lebih efektif.

1.2 Motivasi Penyelidikan dan Pernyataan Masalah

Lantaran kepentingan bidang pengujian ini, para penyelidik perisian sentiasa menjalankan kajian bagi mempertingkatkan proses pengujian ini agar perisian dapat mencapai tahap yang diperlukan dalam masa yang lebih singkat. Dengan implementasi sistem pengautomasian, mereka dapat mencapai tahap pengujian yang lebih efisien dalam masa yang seminimum mungkin secara tidak langsung dapat menjimatkan kos keseluruhan selain memberi jaminan terhadap kualiti perisian terbabit. Walau bagaimanapun, mereka dilihat masih berhadapan dengan

permasalahan untuk mencari kaedah yang terbaik untuk menganalisis, mengukur serta melakukan penilaian berkenaan tahap prestasi ujian yang dilaksanakan terhadap perisian mereka. Salah satu kaedah yang dikenal pasti dapat membantu dalam mengatasi permasalahan mereka adalah melalui analisis liputan kod (Lingampally et al., 2007).

Aplikasi analisis liputan kod ini dimulai dengan membina algoritma liputan untuk disisipkan ke dalam pelaksanaan kod program sewaktu pengujian bagi tujuan memudahkan pengatur cara untuk menilai kod sumber berdasarkan keluaran yang dijana oleh algoritma liputan terbabit. Umumnya, pengukuran liputan kod ini bertujuan membekalkan pengguna dengan informasi mengenai sejauh manakah kes-kes ujian dalam sesuatu pengujian telah menjalankan keseluruhan struktur kod sumber sistem yang diuji. Dengan kata lain, berpandukan kepada rumusan liputan kod yang diperolehi, pengguna akan sentiasa peka akan tahap pengujian yang dijalankan ke atas sistem mereka selain dapat mengukur prestasi sesebuah kes ujian yang digunakan.

Peningkatan mendadak dalam bidang penyelidikan pengujian yang bersandarkan liputan ini adalah disebabkan oleh beberapa faktor. Pertamanya, para penyelidik tidak mempunyai mekanisme yang baik untuk mengetahui sebanyak mana bahagian kod program mereka yang telah dijalankan sewaktu pengujian. Disebabkan oleh arahan dan permintaan daripada golongan atasan yang hanya mahukan laporan mengenai status program mereka pada setiap fasa pengujian dan ditambah pula dengan kekangan jangka masa projek, maka para pengatur cara lebih cenderung menjalankan pengujian ke atas program hanya sekadar untuk melepasi tanda aras

tertentu dalam fasa pengujian seperti pengujian unit, pengujian integrasi atau pengujian sistem tanpa menghiraukan sama ada aliran struktur kawalan program tersebut dijalankan sepenuhnya ataupun tidak (Yang, Li, & Weiss, 2009).

Hal yang demikian menyebabkan mereka sentiasa terikat dengan fasa pengujian yang hanya tertumpu kepada tiga proses berikut iaitu proses pindaan kod bagi tujuan integrasi antara unit, proses reka bentuk integrasi modul sistem serta proses pengujian terhadap sistem yang direka. Kitaran ‘pindaan, reka dan uji’ sebegini adalah tidak efektif kerana terlalu banyak langkah manual digunakan sewaktu kitaran (contohnya pindaan kod yang dilakukan secara manual). Oleh kerana tiada mekanisme khusus untuk menilai kod program ketika dalam kitaran fasa ini maka banyak kecacatan struktur kod terlepas daripada pemeriksaan seterusnya menyebabkan sistem perisian pincang tugas pada fasa yang lebih lanjut.

Keduanya, tanpa prosedur khusus yang mampu menilai tahap kesempurnaan sesebuah pengujian itu, maka para penyelidik sering berada dalam dilema bagi menentukan sama ada proses pengujian yang dijalankan terhadap program mereka sudah cukup sempurna ataupun tidak (Goodenough & Gerhart, 1975; Zhou & Miao, 2010; Zhu, Hall, & May, 1997) kerana mereka sedar bahawa tanpa pengujian yang cukup dan menyeluruh, sistem perisian yang dibangunkan berkemungkinan mendatangkan risiko kepada pengguna sekiranya dipasarkan terutamanya dari segi keselamatan sekiranya program pincang tugas sewaktu dioperasikan oleh pengguna.

Ketiganya, para penyelidik perisian tidak mempunyai kaedah khusus untuk menentukan sama ada kes-kes ujian yang dibina berdasarkan spesifikasi keperluan

pada awalan fasa pembangunan adalah berkualiti dan mencukupi untuk dilaksanakan dalam pengujian dengan sistem yang dibangunkan (Jiang & Chan, 2010; Rajan, 2006; Wloka, Høst, & Ryder, 2010; Zhu, 1995). Hal yang demikian boleh memberikan keluaran pengujian yang tidak tepat tanpa disedari oleh pengatur cara. Kebanyakan pengatur cara beranggapan bahawa dengan kejayaan melaksanakan semua kes ujian yang dibina, maka pengujian ke atas sistem yang dibina telah mencukupi. Namun pada hakikatnya, kes ujian yang terbabit berkemungkinan tidak melaksanakan aliran kawalan sistem dengan sepenuhnya. Aliran kawalan yang tertinggal ini mungkin terdiri daripada kod mati atau *dead code* yang boleh menyebabkan sistem pincang tugas dalam operasi selanjutnya.

Terdorong untuk menyelesaikan permasalahan tersebut, maka alat analisis liputan kod sumber terbuka ini yang dinamakan CovAT ini dihasilkan bagi melaksanakan pengujian bersandarkan liputan secara automatik terhadap struktur kod sumber sistem perisian dan berupaya membantu pengguna untuk menilai sejauh mana kod telah dijalankan oleh kes ujian secara tidak langsung dapat menilai impak pengujian terhadap struktur aliran kawalan kod. CovAT mampu merekodkan maklumat berkenaan pelaksanaan struktur aliran kawalan kod sumber termasuklah elemen unit, fungsi mahupun blok yang hadir di dalam kod sumber program terbabit.

Selain dapat menghasilkan rumusan liputan kod yang baik, ia juga berupaya untuk menyenaraikan secara terperinci maklumat mengenai lokasi elemen kod yang telah dijalankan mahupun tidak oleh kes ujian yang dibekalkan dalam pengujian. Dengan merujuk kepada maklumat terbabit, pengguna dapat merancang dengan lebih terperinci dalam membina kes ujian bagi proses pengujian yang lebih efektif

selanjutnya secara tidak langsung dapat menilai kualiti kes ujian bagi proses sebelumnya (Lee, 2002; Lingampally et al., 2007).

Dengan implementasi yang menyokong teori seni bina terbuka lantaran penerapan reka bentuk bermodul dalam pembangunan alat analisis liputan ini, ia menawarkan dua pilihan kepada pengguna sama ada untuk menjalankan operasi menggunakan sistem sendiri CovAT ataupun dengan menggunakan pustaka yang mengandungi fungsi-fungsi utama CovAT yang boleh dimanipulasikan untuk digunakan oleh program pihak ketiga yang lain melalui fungsi panggilan yang ringkas. Implementasi CovAT dalam domain pustaka ini adalah sangat fleksibel dari segi konsep pengaturcaraan kerana menekankan elemen kebolegunaan semula dan kebolehlanjutan kod yang mana pengatur cara boleh menambah, menatar atau meminda rutin yang terkandung dalam pustaka tersebut mengikut kehendak spesifikasi pengujian mereka.

Hal yang demikian amat berguna kepada pengatur cara yang ingin menggunakan sebahagian rutin pelaksanaan CovAT yang dimuatkan dalam pustaka tersebut dengan perisian pengaturcaraan lain tanpa perlu menulis semula atur cara kod daripada awal. Mereka juga boleh menambah rutin dalam pustaka CovAT bagi melaksanakan pengujian liputan yang lebih teliti ke atas program yang dibangunkan mereka mengikut spesifikasi keperluan masing-masing. Selain itu, pengaturcaraan untuk sistem terbuka seperti ini juga secara tidak langsung menawarkan integrasi yang longgar antara enjin program analisis liputan dengan antara muka pengguna atau GUI sekaligus membuka ruang kepada pengguna untuk mengubahsuai GUI

mengikuti penyelesaian masing-masing tanpa memberi kesan kepada proses utama alat analisis CovAT.

1.3 Matlamat dan Objektif Penyelidikan

Matlamat utama penyelidikan ini adalah untuk menghasilkan sebuah aplikasi yang berupaya untuk menjalankan analisis liputan terhadap kod program dalam pengujian. Bagi mencapai tujuan terbabit, beberapa garis panduan serta objektif telah ditetapkan seperti berikut:

- i) Menyiasat serta mengenal pasti teknik serta metrik terkini yang sesuai untuk analisis liputan kod.
- ii) Menjalankan tinjauan serta penyelidikan ke atas alat-alat analisis liputan sedia ada bagi mengenal pasti ciri-ciri dominan yang penting.
- iii) Membangunkan dan mengesahkan alat analisis liputan kod yang dibina menggunakan kajian-kajian kes yang sesuai.

1.4 Metodologi Penyelidikan

Penyelidikan yang dijalankan adalah berfokus pembangunan alat analisis automatik yang mempunyai kriteria yang mencukupi bagi menjalankan analisis liputan kod. Bagi memenuhi objektif penyelidikan seperti yang dinyatakan, garis panduan metodologi penyelidikan yang perlu dijalankan adalah seperti berikut:

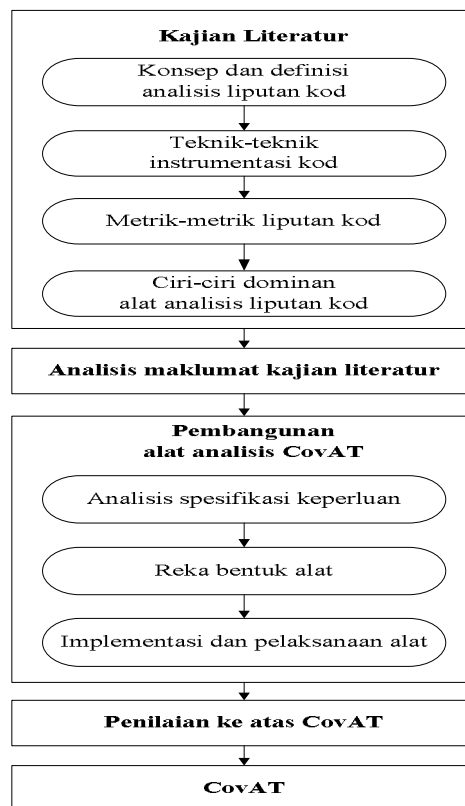
- i. Menjalankan kajian literatur mengenai konsep analisis liputan kod serta beberapa aspek penting analisis liputan kod seperti berikut:

- a. Teknik-teknik instrumentasi kod – kajian dijalankan ke atas beberapa teknik instrumentasi yang lazim digunakan pada masa kini. Kelebihan dan kekurangan untuk setiap teknik dibanding dan disenaraikan bagi memilih teknik terbaik.
 - b. Metrik-metrik liputan kod – metrik liputan kod asas serta variasi metrik turut dinilai bagi menentukan metrik yang dapat memberikan hasil keluaran liputan yang baik dan tepat.
 - c. Ciri-ciri dominan yang penting dalam aplikasi alat analisis liputan – perbandingan dijalankan ke atas alat-alat analisis sedia ada di pasaran sama ada secara komersial ataupun secara percuma bagi menentukan ciri-ciri dominan yang penting dan berguna dalam menjalankan analisis liputan dengan sempurna. Perbandingan juga dibuat untuk melihat spesifikasi keperluan asas yang diperlukan untuk menjalankan analisis liputan dengan berkesan.
- ii. Menganalisis dan menilai maklumat yang diperolehi daripada kajian literatur bagi mengenal pasti spesifikasi keperluan yang sesuai dalam pembangunan alat analisis liputan.
 - iii. Membina dan mengimplentasi reka bentuk sistem berdasarkan spesifikasi keperluan dalam langkah 2. Reka bentuk yang dibina adalah berteraskan konsep bermodul yang menekankan pembinaan sistem secara bermodul yang menerapkan kebolehlanjutan serta kebolehgunaan semula fungsi dalam sistem. Reka bentuk setiap modul dalam sistem juga dibina secara khusus untuk menjalankan analisis liputan kod melalui teknik instrumentasi secara kod sumber yang dapat memberikan keluaran liputan yang baik. Untuk tujuan menyediakan maklumat liputan kepada pengguna, modul-modul sistem juga

direka agar berupaya menjalankan analisis liputan menggunakan metrik liputan unit, fungsi dan blok. Selepas pembangunan alat selesai, pengujian dijalankan untuk mengesan kehadiran pepijat dalam kod.

- iv. Menjalankan penilaian untuk menguji keberkesanan alat analisis yang dibangunkan agar objektif serta matlamat penyelidikan dapat dicapai. Proses pengesahan akan dijalankan ke atas alat yang dibangunkan menggunakan tiga kajian kes iaitu modul-modul sistem CovAT sendiri serta dua program pihak ketiga iaitu program pengesahan segi tiga serta program menu pesanan piza Danillo.

Rajah 1.2 berikut merupakan aliran keseluruhan proses metodologi penyelidikan yang dilaksanakan:



Rajah 1.2 Aliran keseluruhan metodologi penyelidikan

1.5 Organisasi Tesis

Bab 1 sebelum ini menghuraikan secara kasar mengenai cabang pengujian perisian dan matlamat utama pelaksanaannya dituruti dengan sedikit informasi akan kepentingan pengujian berlandaskan liputan serta impaknya kepada pembangunan perisian secara umum. Selebihnya, penulisan tesis ini merangkumi empat lagi bab seperti berikut:

Bab 2 dimulai dengan huraian definisi dan konsep-konsep asas analisis liputan kod seperti instrumentasi kod dan juga mekanisme pengukuran liputan melalui metrik-metrik liputan. Seterusnya, turut diterangkan mengenai teknik-teknik pengujian yang memainkan peranan penting dalam analisis liputan kod khususnya dalam kriteria pemilihan kes ujian untuk pengujian bersandarkan liputan ini. Selain itu, turut dihuraikan secara jelas mengenai aliran proses analisis liputan kod dalam reka bentuk perisian dan kepentingannya dalam menentukan kualiti sesebuah perisian dalam pembangunan. Analisis serta pemerhatian terhadap beberapa alat analisis lain turut disertakan dalam bab ini.

Seterusnya, Bab 3 akan menyentuh secara lengkap konsep reka bentuk serta pelaksanaan aplikasi liputan kod seperti yang diusulkan. Perincian mengenai setiap modul sistem akan dihuraikan satu persatu dengan bantuan gambar rajah bagi memudahkan pemahaman. Implementasi CovAT dalam domain sendiri dan juga secara panggilan domain pustaka turut dihuraikan dalam bab ini.

Dalam Bab 4, penilaian terhadap alat analisis liputan kod yang dibina akan dijalankan melalui beberapa eksperimen menggunakan beberapa kajian kes yang mewakili senario permasalahan harian. Huraian mengenai setiap kajian kes yang dipilih serta objektif untuk setiap eksperimen akan diterangkan secara jelas bagi membuktikan bahawa alat yang dibina mampu beroperasi secara praktikal. Masukan dan keluaran bagi setiap eksperimen akan dipersembahkan dalam bentuk jadual dan graf untuk memudahkan pemahaman. Analisis dan perbincangan mengenai prestasi aplikasi alat analisis liputan yang dibina ini juga turut dimuatkan dalam bab ini.

Akhir sekali, rumusan dan kesimpulan mengenai penyelidikan ini akan disertakan di dalam Bab 5. Prestasi CovAT dalam melaksanakan analisis liputan akan dirumuskan secara keseluruhannya berdasarkan keputusan dalam Bab 4. Beberapa cadangan ke arah penambahbaikan penyelidikan serta aplikasi yang dibina turut diselitkan di bab akhir ini untuk kegunaan di masa hadapan.

BAB 2

KAJIAN LITERATUR

2.1 Pengenalan

Bidang perisian kini melibatkan tahap kerumitan reka bentuk serta saiz sesebuah program yang semakin lama semakin meningkat, tambahan pula dengan peningkatan mendadak dalam saiz kapasiti peranti ingatan untuk menampung keperluan pelaksanaan perisian yang kini boleh didapati dengan kos yang murah. Maka, penyelidikan bagi mencari kaedah pengujian terbaik yang amat kritikal daripada sudut produktiviti dan kualiti perisian dilihat semakin lama semakin kompleks dan rumit serta menelan kos yang tinggi selain memakan masa yang lama.

Secara praktikalnya, untuk menjalankan pengujian secara menyeluruh atau tuntas adalah mustahil kerana proses untuk mengenal pasti setiap permutasi input pengujian yang mungkin atau lebih dikenali sebagai 'kes ujian' ini boleh menjangkau masa yang agak lama dan pengatur cara mestilah mempunyai tahap kefahaman yang tinggi mengenai struktur aliran kawalan kod program yang diuji. Hal ini adalah sukar dilakukan dalam kekangan jangka masa yang terhad lebih-lebih lagi sekiranya program yang diuji merupakan program pihak ketiga yang bukan dihasilkan sendiri ataupun dalam senario sesebuah program yang mempunyai tahap kekompleksan aliran kawalan kod yang tinggi.

Lantaran kesedaran mengenai kepentingan pengujian yang mencukupi dalam pembangunan sesebuah perisian, maka pelbagai variasi pendekatan pengujian digunakan untuk membantu para pengatur cara dalam menentusahkan program mereka agar bebas daripada kecacatan akibat pepijat dalam kekangan jangka masa serta kos projek. Oleh yang demikian, sebagai alternatif kepada pengujian tuntas ini banyak kaedah baru khususnya dalam strategi peminimuman kes ujian telah diperkenalkan antaranya pengujian tatasusunan ortogon (Popovic & Lazic, 2009), pengujian cara berpasangan atau *pairwise testing* (Tai & Lei, 2002) dan pengujian T-hala (Zamli et al., 2009). Satu lagi kaedah pengujian yang diguna pakai secara meluas untuk memantau kriteria tersebut adalah melalui pengujian bersandarkan liputan atau *coverage-based testing*.

Huraian dalam bab ini dipecahkan kepada lima bahagian utama. Bahagian pertama merangkumi huraian secara lanjut mengenai analisis liputan kod berserta konsep asas pelaksanaan analisis liputan kod seperti proses instrumentasi, pemilihan metrik liputan dan teknik-teknik pengujian yang berkaitan. Bahagian kedua pula menerangkan dengan jelas fasa pelaksanaan analisis liputan kod dalam reka bentuk perisian secara umum diikuti dengan bahagian ketiga yang menghuraikan aliran proses analisis liputan kod secara terperinci. Bahagian keempat menegaskan kepentingan analisis liputan serta impaknya terhadap pengujian perisian secara keseluruhan. Bahagian kelima dan terakhir akan meliputi analisis serta pemerhatian yang telah dijalankan ke atas beberapa alat analisis liputan yang lain.

2.2 Analisis Liputan Kod

Matlamat utama pengujian liputan adalah untuk membantu pengatur cara untuk memastikan setiap aliran kawalan struktur program yang dibina dilaksanakan oleh kes ujian sewaktu proses pengujian berlangsung. Hal yang demikian adalah untuk mengesahkan bahawa sistem perisian bebas daripada pepijat yang menyebabkan sistem perisian beroperasi di luar jangkaan sekaligus memastikan program menepati spesifikasi yang telah ditetapkan pada fasa awal dalam pembangunan perisian.

Dengan kata lain, analisis liputan kod ini boleh dianggap sebagai satu kaedah nyahpepijat yang lebih moden dan terperinci. Dengan penyepaduan analisis liputan kod dengan sistem automasi, ia boleh dijadikan alat sokongan pengujian yang amat bermanfaat dari segi proses penyahpepijatan dalam membantu pengatur cara untuk mengenal pasti permasalahan yang hadir dalam binaan kod mereka. Secara ringkas, berikut merupakan tujuan asal analisis liputan kod diperkenalkan sebagai salah satu mekanisme pengujian perisian (Jiang & Chan, 2010; Zhu et al., 1997):

- i. Mengenal pasti bahagian dalam struktur aliran kawalan kod yang tidak dilaksanakan oleh kes-kes ujian.
- ii. Menentukan sama ada kes ujian perlu ditambah untuk meningkatkan keluaran liputan sekaligus mengukur kualiti kes ujian itu sendiri.
- iii. Mengesan kes ujian lewah yang tidak meningkatkan keluaran liputan.
- iv. Memberikan keluaran liputan kod dalam bentuk nilai pengukuran kuantitatif yang boleh diguna pakai bagi menggambarkan kualiti sesebuah perisian.

Analisis liputan kod ini mula diperkenalkan sekitar tahun 1980-an ketika bahasa pengantaraan komputer generasi ke-3 seperti Pascal, ADA dan C digunakan secara meluas dalam pengaturcaraan. Bahasa pengantaraan tersebut mengamalkan pengaturcaraan berstruktur yang melatih pengatur cara supaya lebih berdisiplin dan teratur dalam menulis kod mereka. Sebagai contoh mudah, sekiranya mereka ingin menggunakan pemboleh ubah tertentu dalam kod mereka, ia mesti diisytiharkan terlebih dahulu dengan jenis maklumat yang hendak digunakan sama ada integer, perduaan, rentetan aksara dan sebagainya. Secara tidak langsung, pendekatan sebegini membolehkan pengkompil program mengenal pasti kesalahan sekiranya pengatur cara tersilap mewakili jenis maklumat kepada pemboleh ubah terbabit yang tidak bertepatan dengan penggunaannya dalam program tersebut.

Namun begitu, pengaturcaraan berstruktur sebegini hanya memberi kelebihan untuk pengesanan pepijat dalam perwakilan data manakala permasalahan dalam perilaku logik sesebuah program tidak dapat dikenal pasti dengan jelas oleh pemproses program. Sebagai contoh, satu program yang pincang tugas akibat alirannya terperangkap dalam operasi gelung tak terhingga yang mungkin disebabkan oleh pemboleh ubah tertentu yang tidak ditokok kiraannya, lalu gagal memenuhi ujian logik dan aritmetik bagi program terbabit. Untuk mengesan kewujudan ralat logik seperti ini secara manual adalah memakan masa lebih-lebih lagi jika program tersebut mengandungi ratusan baris kod.

Atas dasar untuk menyelesaikan permasalahan tersebut, maka analisis liputan kod diperkenalkan bagi membantu pengatur cara mengenal pasti permasalahan logik program dengan mudah. Pada awal perkenalannya, analisis liputan kod hanya dapat