

SKEMA PENJANAAN DATA DAN PEMBERAT UJIAN BERASASKAN KAEDAH ANALISIS NILAI SEMPADAN

ZARINA SHUKUR^{1*}, ROHAIDA ROMLI² & ABDUL RAZAK HAMDAN³

Abstrak. Kertas kerja ini menjelaskan reka bentuk satu skema bagi penjaanaan data dan pemberat ujian untuk memarkahkan aturcara pelajar. Penjaanaan data ujian dibuat berdasarkan teknik analisis-nilai sempadan (penguajian kotak-hitam). Nilai pemberat akan diberikan untuk setiap data ujian berdasarkan kepada nilai kritikal bagi data ujian tersebut. Nilai pemberat yang diberi digunakan dalam pemberian markah semasa penaksiran aturcara. Skema ini telah digunakan sebagai komponen dalam Sistem Penaksiran Aturcara C, STaksirC.

Kata kunci: Data ujian, analisis nilai sempadan, pemberat ujian, aturcara C, teknik pemarkahan

Abstract. This paper describes the design of a scheme to generate test data and test weight for marking student's program. The generation of test data was based on boundary value analysis technique. Each test datum will be given a weight based on how critical the data is. The given weight was used during program assessment. This scheme has been used as one of the components in C Evaluation System, also known as STaksirC.

Keywords: Test data, boundary value analysis, test weight, C program, marking technique

1.0 PENDAHULUAN

Terdapat pelbagai usaha ke arah pengautomasian pemarkahan bagi pelbagai bentuk tugas seperti pemarkahan pelbagai pilihan, pengiraan, pengaturcaraan komputer, teks/esei, grafik dan pelbagai bentuk lagi. Fokus kertas ini ialah berkaitan dengan pemarkahan berasaskan aturcara. Pemarkahan aturcara berautomasi kebiasaannya melibatkan penguajian. Pemarkahan aturcara berasaskan penguajian memerlukan suatu set data ujian yang akan digunakan untuk melaksanakan kes ujian. Data ujian adalah subset data input yang kecil yang mewakili kemungkinan domain input yang sedia ada bagi sesuatu perisian [1] ataupun secara ringkasnya dikenali sebagai data input aturcara. Kes ujian adalah keadaan situasi input tertentu yang hendak diuji yang mengandungi suatu set data ujian [2]. Kes ujian direka bentuk untuk menguji aturcara dengan menggunakan input dan output yang betul dan yang salah. Tujuannya adalah untuk mengatasi sebanyak mungkin ralat aturcara. Penjaanaan data ujian pula adalah proses menghasilkan data input aturcara yang memenuhi kriteria penguajian [3].

^{1,2&3}Fakulti Teknologi dan Sains Maklumat, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor.

* Penulis utama: Tel: 03-89216720, Faks: 03-89256732, Email: zs@ftsm.ukm.my

Bagi membolehkan sesuatu pengujian berautomasi ke atas aturcara dilakukan, isu berikut perlu dipertimbangkan [4]:

- (i) keperluan kepada data ujian dan teknik memilihnya
- (ii) keperluan kepada jawapan guru
- (iii) keperluan kepada teknik penghasilan markah

Sehingga kini, kajian dalam pengautomasian penjanaan data ujian banyak dilakukan oleh kumpulan penyelidik dengan fokus yang berbeza. Manakala isu pemberian markah pula amat kurang diberi perhatian. Kertas ini mengesyorkan skema pemberian markah berdasarkan pemberat dengan mengambil kira tahap kritikal sesuatu kes ujian. Sehubungan itu, permasalahan penjanaan data ujian berasaskan tahap kritikal adalah antara yang dibincangkan dalam kertas ini.

2.0 PENGUJIAN

Secara amnya, pengujian perisian boleh dibahagikan kepada dua jenis, iaitu pengujian statik dan pengujian dinamik [5]. Pengujian statik terdiri daripada pengujian sintaks dan semantik. Manakala pengujian dinamik boleh dibahagikan kepada tiga jenis iaitu: pengujian kotak hitam, pengujian kotak putih dan pengujian regresi. Perbandingan antara kesemua teknik pengujian ini ditunjukkan dalam Jadual 1.

Jadual 1 Perbandingan teknik pengujian perisian

Ciri	Pengujian statik		Pengujian dinamik		
	Sintaks	Semantik	Kotak hitam	Kotak putih	Regresi
Keperluan pelaksanaan perisian	×	×	√	√	√
Keperluan input	×	×	√	√	√
Keperluan pemeriksaan kod sumber	√	×	×	√	√
Kesesuaian perisian	Perisian yang tiada sumbangan input	Perisian yang tiada sumbangan input	Perisian yang hanya diminati akan kebetulsalahannya	Perisian yang tidak terlalu kompleks	Perisian yang alami perubahan kod sumber

Seperti yang ditunjukkan dalam Jadual 1, pengujian dinamik kotak hitam sesuai jika hanya kebetulsalahan aturcara ingin disemak. Bagi suatu sistem pemarkahan automatik untuk tugas aturcara penuntut awalan, maka penyemakan ini ditumpukan.

Pengujian dinamik memerlukan data ujian yang digunakan dalam kes ujian untuk menyemak perisian dalam memperolehi sebanyak mungkin ralat. Oleh itu, pemilihan data ujian perlu dilakukan berasaskan suatu teknik.

Pemilihan data ujian juga disebut reka bentuk data ujian. Bagi pengujian kotak hitam [6, 7], antara teknik reka bentuk tersebut ialah; Pembahagian Setara, Analisis Nilai Sempadan, Pengujian Rawak dan Pengujian berasaskan Spesifikasi. Antara keempat-empat teknik reka bentuk tersebut, kaedah Analisis Nilai Sempadan mengklasifikasi data ujian berasaskan sempadan dan ini memberi laluan kepada pengkaji meletakkan nilai pemberat kepada data ujian tersebut. Walaupun kaedah Analisis Nilai Sempadan digunakan, maklumat asas yang diperlukan oleh teknik ini adalah dirujuk sepenuhnya dari spesifikasi aturcara (dalam kes ini soalan yang dikemukakan).

Setelah teknik pemilihan data ujian ditentukan, data ujian tersebut boleh disediakan sama ada secara manual, salinan daripada versi lama atau dijanakan secara automatik oleh peralatan perisian. Peningkatan dalam bilangan produk perisian dan kekompleksannya menjadikan dua cara penyediaan data ujian yang pertama yang disebutkan kurang sesuai digunakan. Penjanaan data ujian secara automatik diperlukan untuk menyokong pengujian perisian dengan lebih berkesan [8].

Seni bina penjana data ujian automatik dapat diklasifikasi berdasarkan paradigma penjanaan. Paradigma penjanaan ini pula berasaskan kepada reka bentuk data ujian. Terdapat beberapa paradigma dalam penjanaan data ujian automatik. Berasaskan paradigma ini, pelbagai penjana data ujian dihasilkan. Jadual 2 menunjukkan

Jadual 2 Kelas paradigma penjanaan data ujian

Pengujian dinamik	Paradigma penjanaan	Reka bentuk data ujian	Contoh penjana
Kotak hitam	Penjanaan data ujian rawak [9]	Secara rawak	-
Kotak hitam	Penjanaan data ujian simbolik	Berdasarkan kriteria kecukupan	-
Kotak hitam	Penjanaan data ujian berasaskan spesifikasi	Berasaskan spesifikasi data	-
Kotak putih	Penjanaan data ujian dinamik [10]	Berasaskan anggapan bahawa aturcara adalah fungsi-fungsi	TESTGEN [9] ADTEST [11]
Kotak putih	Penjanaan data ujian berasaskan Alkhawarizmi genetik [12]	Lanjutan secara rawak	GADGET [13]
Kotak putih	Penjanaan data ujian lintasan wayar	Berasaskan domain input	-
Kotak putih	Penjanaan data ujian berorientasikan matlamat dan pendekatan berantai	Lanjutan penjanaan data ujian dinamik	-

sebahagian penjana yang dihasilkan berdasarkan paradigma yang dinyatakan dengan memfokus kepada pengujian dinamik.

Kajian yang dibincangkan dalam kertas ini ialah menggunakan teknik pengujian dinamik secara kotak hitam bagi menjana data ujian untuk tujuan pemarkahan aturcara pelajar. Reka bentuk data ujian yang digunakan ialah kaedah Analisis Nilai Sempadan yang akan dibincangkan dalam Seksyen 3.0. Seterusnya, skema penjanaan data ujian yang berasaskan teknik ini akan dibincangkan dalam Seksyen 4. Seperti yang telah dinyatakan terlebih dahulu, kajian ini bukan sahaja bertujuan menjana data ujian tetapi yang utamanya turut menghasilkan pemberat bagi data tersebut untuk tujuan penentuan sukatan untuk penaksiran kebetulsalahan aturcara. Sehubungan itu, kaedah pemberian nilai pemberat kepada setiap data ujian (kes ujian) akan dijelaskan dalam Seksyen 5. Skema penghasilan data ujian dan pemberat ujian ini telah diimplementasi dalam Sistem Penaksiran Aturcara C, STaksirC [14].

3.0 KAEDAH ANALISIS NILAI-SEMPADAN

Analisis nilai sempadan menganggap data ujian yang berada hampir dengan sempadan-sempadan domain input mempunyai kejituan yang tinggi berbanding dengan data ujian yang berada jauh dari sempadan tersebut [15]. Menurut laporan eksekutif IPL Information Processing [16], analisis nilai-sempadan menggabungkan darjah pengujian negatif dalam reka bentuk ujian yang dilaksanakan. Ia menganggap ralat-ralat akan wujud pada atau berhampiran sempadan-sempadan domain input.

Contoh Penggunaan

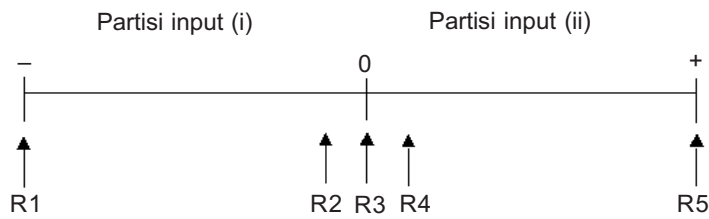
Bagi menjelaskan lagi konsep analisis nilai-sempadan ini, penerangan melalui contoh soalan aturcara akan digunakan. Sila rujuk contoh spesifikasi fungsian bagi soalan aturcara di bawah.

Soalan 1 :
Buat aturcara untuk mengira punca kuasa dua bagi nombor nyata.

Spesifikasi fungsian :

Input : nombor nyata
Output : nombor nyata
Proses : Apabila diberi input 0 atau lebih lebih besar dari 0, punca kuasa dua akan dipulangkan. Apabila diberi input kurang dari 0, mesej "Ralat, input tidak sah" dipaparkan dan nilai 0 (sifar) dipulangkan.

Berdasarkan spesifikasi fungsian yang diberikan di atas, kes-kes ujian akan direka bentuk untuk menguji aturcara sama ada berada pada atau di luar nilai-nilai sempadan domain input. Spesifikasi fungsian ini terdiri daripada maklumat input, output dan proses yang dijangkakan berlaku. Reka bentuk kes ujian untuk contoh soalan ini



Rajah 1 Sempadan-sempadan partisi input kes ujian

akan dijelaskan berdasarkan pendekatan yang telah dibincangkan dalam laporan eksekutif IPL Information Processing [16]. Dua partisi input diambil kira dalam contoh soalan di atas, iaitu partisi input (i) dan partisi input (ii). Dua partisi ini dihasilkan berdasarkan pernyataan proses bagi spesifikasi fungsian contoh soalan di atas. Sempadan-sempadan partisi input bagi contoh ini digambarkan oleh Rajah 1.

Berdasarkan Rajah 1 tersebut, partisi input (i) berada di antara sempadan 0 dan sempadan nombor nyata paling negatif. Manakala partisi input (ii) berada di antara sempadan 0 dan sempadan nombor nyata paling positif. Kedua-dua partisi input berkongsi sempadan 0 yang sama. 5 kes ujian boleh dipilih berdasarkan sempadan ini iaitu R1, R2, R3, R4 dan R5, dengan perincian seperti yang ditunjukkan dalam Jadual 3.

Jadual 3 Jadual kes ujian untuk contoh soalan 1

Kes ujian	Partisi input	Input	Output	Komen
R1	Sempadan bawah partisi (i)	Nombor nyata yang lebih negatif ($0 - \infty$)	0 (sifar)	Ralat, input tidak sah
R2	Sempadan atas partisi (i)	Nombor nyata yang kecil sedikit daripada sifar ($0 - \delta$)	0 (sifar)	Ralat, input tidak sah
R3	Di luar sempadan atas partisi (i) dan sempadan bawah partisi (ii)	Nilai sifar (0)	0 (sifar)	Input sah
R4	Sempadan bawah partisi (ii)	Nombor nyata yang lebih besar sedikit daripada sifar ($0 + \delta$)	Punca kuasa dua positif	Input sah
R5	Sempadan atas partisi (ii)	Nombor nyata yang lebih positif ($0 + \infty$)	Punca kuasa dua positif	Input sah

Berdasarkan perbincangan kepada konsep analisis nilai-sempadan ini, secara umumnya didapati bahawa sebelum kes ujian direka bentuk, spesifikasi fungsian soalan aturcara perlu ditentukan terlebih dahulu bagi menentukan sempadan-sempadan domain input aturcara. Daripada nilai sempadan-sempadan domain input yang telah diperolehi, kes-kes ujian boleh direka bentuk pada keadaan berikut:

- (i) Pada sempadan-sempadan domain input,
 - (ii) Bersebelahan dengan sempadan-sempadan domain input dengan perubahan yang kecil (nilai sempadan domain input $+ \delta$ dan nilai sempadan domain input $- \delta$), dan
 - (iii) Bersebelahan dengan sempadan-sempadan domain input dengan perubahan yang besar (nilai sempadan domain input $+ \infty$ dan nilai sempadan domain input $- \infty$).
- (δ : mewakili perubahan kecil, ∞ : mewakili perubahan besar)

4.0 SKEMA PENJANAAN DATA UJIAN

Data ujian yang dijana bagi setiap aturcara bergantung kepada syarat pembolehubah input dan pembolehubah output bagi aturcara berkenaan. Set data ujian yang dijana bergantung kepada bilangan pembolehubah input yang terlibat dalam setiap aturcara. Bagi kajian ini, tiga parameter berikut akan diambil kira daripada pembolehubah input yang diuji:

- (i) Nama bagi setiap pembolehubah input yang terlibat (rentetan).
- (ii) Jenis data setiap pembolehubah input yang terlibat, iaitu sama ada aksara atau numerik (integer atau nombor nyata).
- (iii) Sifat bagi nilai setiap pembolehubah input yang terlibat, iaitu sama ada bersifat positif, negatif atau global. Pembolehubah yang mengambil hanya data tunggal positif seperti 24, adalah disebut nilainya bersifat positif; manakala bagi pembolehubah yang hanya mengambil nilai -2 pula adalah bersifat negatif; dan pembolehubah yang boleh mengambil sebarang nilai data yang berada dalam suatu julat seperti antara -3 hingga 10 , adalah pembolehubah yang nilainya bersifat global. Apabila sifat nilai pembolehubah adalah global, nilai julat minimum dan julat maksimum perlu ditentukan.

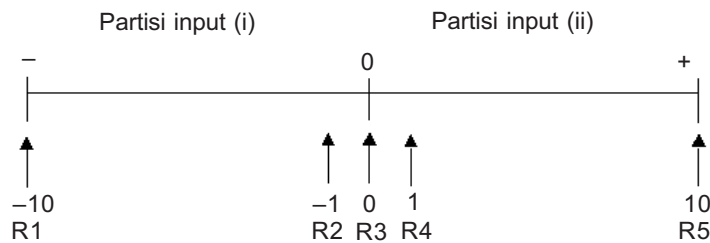
Bagi pembolehubah output, hanya parameter ke (ii) dan ke (iii) dipertimbangkan. Nilai-nilai parameter di atas ditentukan oleh pengajar. Nilai-nilai ini akan mewakili spesifikasi aturcara pengajar iaitu model jawapan bagi soalan aturcara berkaitan. Daripada maklumat spesifikasi aturcara pengajar ini, skema penjanaan data ujian dan sukatan kebetulsalahan aturcara dapat ditentukan. Skop kajian ini hanya mempertimbangkan jenis data numerik kerana ianya bersesuaian dengan kaedah Analisis Nilai Sempadan.

Skema penjanaan data ujian secara automatik ini dibahagikan berdasarkan kriteria jenis pembolehubah numerik (iaitu sama ada integer dan nombor nyata) dan sifat nilai pembolehubah numerik berkenaan (iaitu sama ada bernilai positif, negatif atau global). Pembahagian skema tersebut adalah seperti berikut:

- (i) Numerik integer dan positif
- (ii) Numerik nyata dan positif
- (iii) Numerik integer dan negatif
- (iv) Numerik nyata dan negatif
- (v) Numerik integer dan global
- (vi) Numerik nyata dan global

(a) Pembolehkan numerik (integer) bernilai positif

Skema penjanaan data ujian bagi pembolehkan numerik (integer) bernilai positif digambarkan oleh Rajah 2.



Rajah 2 Skema penjanaan data ujian untuk pembolehkan numerik (integer) bernilai positif

Bagi pembolehkan numerik bernilai positif, nilai sempadan domain inputnya adalah pada sempadan bernilai 0 (sifar). Bagi kes ujian R1 (mewakili $0 - \infty$) dan R5 (mewakili $0 + \infty$), nilai bagi ∞ yang dipilih adalah 10. Ini adalah kerana, walaupun nilai yang lebih besar dipilih, ia tidak memberikan kesan yang berbeza untuk kes ujian berkenaan. Manakala bagi kes ujian R2 (mewakili $0 - \delta$) dan R4 (mewakili $0 + \delta$), nilai bagi δ yang dipilih adalah 1. Ini adalah kerana, walaupun nilai yang lebih kecil dipilih ia tidak memberikan kesan yang berbeza untuk kes ujian berkenaan.

Untuk menjelaskan skema bagi Rajah 2, contoh soalan aturcara 2 akan digunakan:

Soalan 2 :

Buat aturcara yang dapat membaca nilai umur seseorang dan cetak nilai umur berkenaan.

Spesifikasi fungsian :

Input : nilai umur berjenis integer positif

Output : nilai umur berjenis integer positif

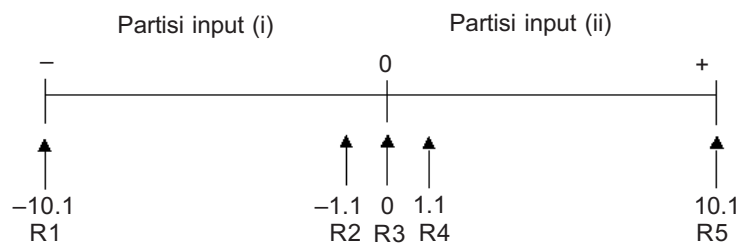
Proses : Apabila diberi input umur 0 atau lebih besar dari 0, nilai umur akan dipulangkan. Apabila diberi input kurang dari 0, mesej "Ralat input tidak sah" dipaparkan dan nilai 0 (sifar) dipulangkan.

Kes ujian yang direka bentuk:

Kes ujian	Partisi input	Data ujian	Output	Komen
R1	Sempadan bawah partisi (i)	-10	0	Ralat, input tidak sah
R2	Sempadan atas partisi (i)	-1	0	Ralat, input tidak sah
R3	Di luar sempadan atas partisi (i) dan sempadan bawah partisi (ii)	0	0	Input sah
R4	Sempadan bawah partisi (ii)	1	1	Input sah
R5	Sempadan atas partisi (ii)	10	10	Input sah

(b) Pembolehkan numerik (nombor nyata) bernilai positif

Skema penjanaan data ujian bagi pembolehkan numerik (nombor nyata) bernilai positif digambarkan oleh Rajah 3.



Rajah 3 Skema penjanaan data ujian untuk pembolehkan numerik (nombor nyata) bernilai positif

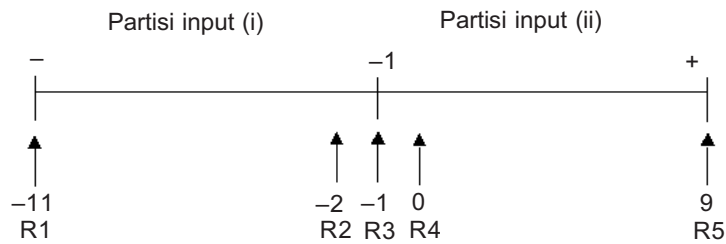
Bagi pembolehkan numerik (nombor nyata) bernilai positif, nilai sempadan domain inputnya adalah pada sempadan bernilai 0 (sifar). Perbezaan antara skema yang melibatkan nombor nyata ini dengan skema yang dibincangkan sebelum ini iaitu nombor integer ialah dari aspek pemilihan nilai δ dan ∞ . Bagi skema ini, masing-masing δ dan ∞ yang dipilih ialah 1.1 dan 10.1.

(c) Pembolehkan numerik (integer) bernilai negatif

Skema penjanaan data ujian untuk pembolehkan numerik (integer) bernilai negatif adalah digambarkan oleh Rajah 4.

Bagi pembolehkan numerik (integer) bernilai negatif, nilai sempadan domain inputnya adalah pada sempadan bernilai -1. Alkhawarizmi penjanaan data ujian bagi skema yang ditunjukkan dalam Rajah 4 adalah sama seperti alkhawarizmi penjanaan data ujian bagi pembolehkan numerik (integer) bernilai positif (rujuk Rajah 2). Cuma yang membezakan kedua-duanya adalah nilai sempadan domain inputnya.

Untuk menjelaskan skema bagi Rajah 4, contoh soalan aturcara berikut akan digunakan:



Rajah 4 Skema penjanaan data ujian untuk pembolehubah berjenis numerik (integer) bernilai negatif

Soalan 3 :

Buat aturcara yang dapat membaca satu nombor integer yang lebih kecil dari sifar dan cetak nombor berkenaan.

Spesifikasi fungsian :

Input: nombor integer

Output: nombor integer

Proses: Apabila diberi input -1 atau lebih kecil dari -1 , nilai nombor negatif akan dipulangkan. Apabila diberi input 0 atau lebih besar dari 0 , mesej "Ralat input tidak sah" dipaparkan dan nilai 0 (sifar) dipulangkan.

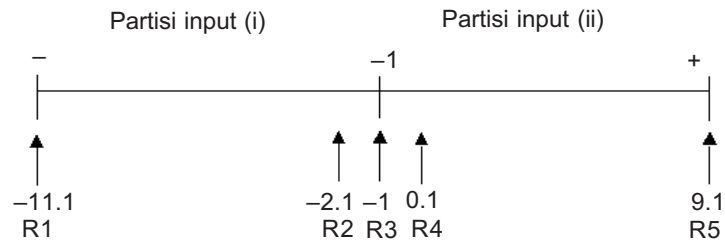
Kes ujian yang direka bentuk:

Kes ujian	Partisi input	Data ujian	Output	Komen
R1	Sempadan bawah partisi (i)	-11	-11	Input sah
R2	Sempadan atas partisi (i)	-2	-2	Input sah
R3	Di luar sempadan atas partisi (i) dan sempadan bawah partisi (ii)	-1	-1	Input sah
R4	Sempadan bawah partisi (ii)	1	0	Ralat, input tidak sah
R5	Sempadan atas partisi (ii)	9	0	Ralat, input tidak sah

(d) Pembolehubah numerik (nombor nyata) bernilai negatif

Skema penjanaan data ujian untuk pembolehubah numerik (nombor nyata) bernilai negatif digambarkan oleh Rajah 5.

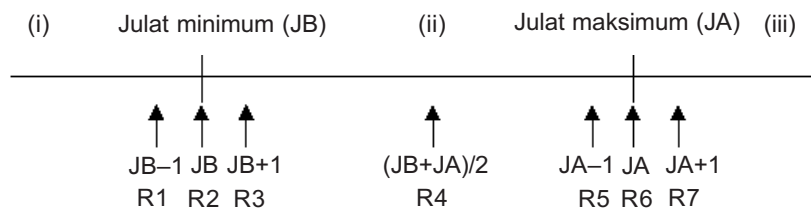
Bagi pembolehubah numerik (nombor nyata) bernilai negatif, nilai sempadan domain inputnya adalah pada sempadan bernilai -1 . Seperti skema (b), skema (d) berbeza daripada skema (c) dari aspek pemilihan nilai δ dan ∞ .



Rajah 5 Skema penjaanaan data ujian untuk pembolehubah berjenis numerik (nombor nyata) bernilai negatif

(e) Pembolehubah numerik (integer) bernilai global

Skema penjaanaan data ujian untuk pembolehubah numerik (integer) bernilai global digambarkan oleh Rajah 6.



Rajah 6 Skema penjaanaan data ujian untuk pembolehubah numerik (integer) bernilai global

Bagi pembolehubah numerik (integer) berjenis global, nilai sempadan domain inputnya adalah pada sempadan bernilai julat minimum (JB) dan pada sempadan bernilai julat maksimum (JA). Untuk menjelaskan skema bagi Rajah 6, contoh soalan aturcara berikut akan digunakan:

Soalan 4 :

Buat aturcara yang dapat membaca satu nombor integer yang mempunyai syarat berikut : nombor ≥ 5 dan nombor ≤ 10 . Kemudian cetak nombor berkenaan.

Spesifikasi fungsian:

Input: nombor integer

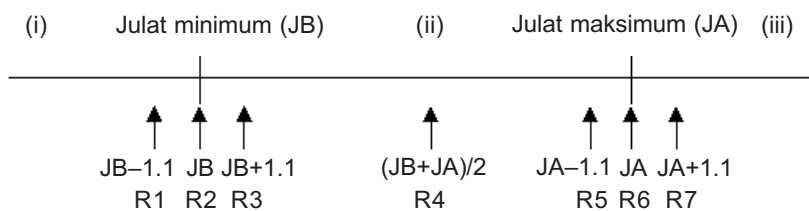
Output: nombor integer

Proses: Apabila diberi input bernilai 5 hingga 10, nilai nombor akan dipulangkan. Apabila diberi input bernilai < 5 atau input bernilai > 10 , mesej "Ralat input tidak sah" dipaparkan dan nilai 0 (sifar) dipulangkan.

Kes ujian yang direka bentuk:

Kes ujian	Partisi input	Data ujian	Output	Komen
R1	Sempadan bawah partisi (i)	4	0	Ralat, input tidak sah
R2	Di luar sempadan atas partisi (i) dan sempadan bawah partisi (ii)	5	5	Input sah
R3	Sempadan bawah partisi (ii)	6	6	Input sah
R4	Dalam sempadan partisi (ii)	8	8	Input sah
R5	Sempadan atas partisi (ii)	9	9	Input sah
R6	Di luar sempadan atas partisi (ii) dan sempadan bawah partisi (iii)	10	10	Input sah
R7	Sempadan bawah partisi (iii)	11	0	Ralat, input tidak sah

(f) Pembolehkan numerik (nombor nyata) bernilai global
 Skema penjanaan data ujian untuk pembolehkan numerik (nombor nyata) bernilai global digambarkan oleh Rajah 7.



Rajah 7 Skema penjanaan data ujian untuk pembolehkan numerik (nombor nyata) bernilai global

Bagi pembolehkan numerik (nombor nyata) berjenis global, nilai sempadan domain inputnya adalah pada sempadan bernilai julat minimum (JB) dan pada sempadan bernilai julat maksimum (JA).

Skema penjanaan data ujian yang dibincangkan di atas merujuk kepada satu pembolehkan input sahaja. Sekiranya aturcara melibatkan lebih daripada satu pembolehkan input, kaedah gabungan akan digunakan untuk menggabungkan data-data ujian yang telah dijana untuk setiap pembolehkan input yang terlibat. Kaedah gabungan ini akan dijelaskan seperti berikut.

4.1 Kaedah Penggabungan Data-data Ujian

Gabungan didefinisikan sebagai [17]:

“Kombinasi n objek yang mengambil r pada suatu masa (dikenali sebagai r -kombinasi bagi n objek) adalah pemilihan tidak tersusun bagi r bagi objek berkenaan.” Di mana n ialah bilangan objek dan r adalah bilangan objek yang diambil dari n objek pada satu masa.

Notasi gabungan adalah: ${}_n C_r = \frac{n!}{r!(n-r)!}$

di mana tanda ‘!’ adalah simbol faktorial yang bermaksud:

$$n! = n.(n-1).\dots.(n-(n-2)).(n-(n-1))$$

Jika r pilihan dibuat beberapa kali dari kumpulan objek yang berbeza dengan saiz yang berbeza (n_1, n_2, \dots, n_i), maka gabungan pilihan yang boleh dibuat ialah:

$$({}_{n_1} C_r) \cdot ({}_{n_2} C_r) \cdot ({}_{n_3} C_r) \cdot \dots \cdot ({}_{n_i} C_r)$$

4.2 Proses Gabungan Data-data Ujian

Bilangan kombinasi data ujian yang terhasil bagi setiap soalan tugas bergantung kepada bilangan pembolehubah input yang terlibat. Sekiranya dua pembolehubah input terlibat, maka proses gabungan data ujian akan melibatkan gabungan antara data-data ujian untuk kedua-dua pembolehubah input berkenaan.

Merujuk kepada formula gabungan yang telah dinyatakan, n dan r akan mewakili nilai-nilai berikut:

- n_i = bilangan data ujian yang dijana untuk pembolehubah input i .
- r = bilangan data ujian yang dipilih daripada koleksi data ujian untuk setiap pembolehubah input pada satu masa.

Bagi kajian ini, r sentiasa bernilai 1 kerana hanya satu data ujian akan diambil dari koleksi data ujian bagi setiap pembolehubah input untuk digabungkan antara satu sama lain. Sebagai contoh:

- (i) Sekiranya bilangan pembolehubah input adalah 2 dengan bilangan data ujian ialah n_1 dan n_2 masing-masing, maka bilangan kombinasi data ujian ialah ${}_{n_1} C_1 \cdot {}_{n_2} C_1$
- (ii) Sekiranya bilangan pembolehubah input adalah 3 dengan bilangan data ujian ialah n_1, n_2 dan n_3 masing-masing, maka bilangan kombinasi data ujian ialah ${}_{n_1} C_1 \cdot {}_{n_2} C_1 \cdot {}_{n_3} C_1$ dan seterusnya mengikut bilangan pembolehubah input.

Secara amnya, bilangan kombinasi data ujian yang terhasil untuk P bilangan pembolehubah input adalah seperti berikut:

$$\text{Bilangan kombinasi data ujian} = \prod_{i=1}^P n_i C_1$$

di mana, P = bilangan pembolehubah input
 n_i = bilangan data ujian untuk pembolehubah input ke i

Bagi menjelaskan lagi kaedah penggabungan data ujian ini, penerangan melalui contoh akan diberikan. Sila rujuk contoh berikut:

Andaikan:

X adalah pembolehubah input pertama dan Y adalah pembolehubah input kedua.

Data ujian dijana untuk X adalah:

Kes ujian	Data ujian
R1	-10
R2	-1
R3	0
R4	1
R5	10

Data ujian dijana untuk Y adalah:

Kes ujian	Data ujian
R1	-10
R2	-1
R3	0
R4	1
R5	10

Rajah 8 menunjukkan kombinasi yang terhasil antara X dan Y. Daripada pengiraan melalui formula gabungan, bilangan kombinasi data ujian untuk X dan Y adalah:

Formula gabungan : ${}_n C_r$

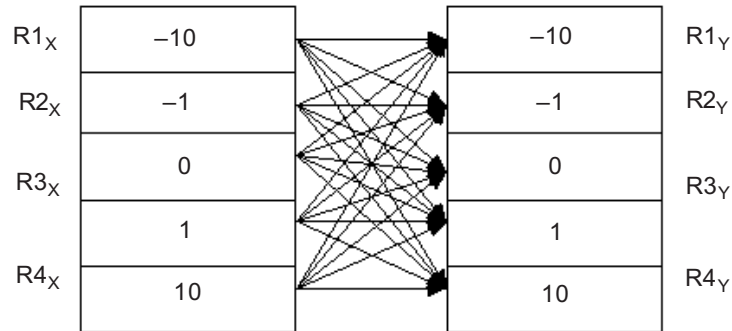
Didapati nilai $n = 5$ dan $r = 1$;

Oleh itu, bilangan kombinasi data ujian yang boleh dibentuk adalah:

$$\begin{aligned} {}_5 C_{1.5} C_1 &= \frac{5!}{1!(5-1)!} \times \frac{5!}{1!(5-1)!} \\ &= 120 / 24 \times 120 / 24 \\ &= 5 \times 5 \\ &= 25 \text{ kombinasi data ujian} \end{aligned}$$

Senarai 25 kombinasi data ujian yang terhasil dari (X, Y) adalah:

$(-10,-10), (-10,-1), (-10,0), (-10,1), (-10,10)$
 $(-1,-10), (-1,-1), (-1,0), (-1,1), (-1,10)$
 $(0,-10), (0,-1), (0,0), (0,1), (0,10)$
 $(1,-10), (1,-1), (1,0), (1,1), (1,10)$
 $(10,-10), (10,-1), (10,0), (10,1), (10,10)$



Rajah 8 Kombinasi kes ujian bagi dua pembolehubah

Daripada perbincangan sub topik ini, didapati bahawa kaedah gabungan diperlukan untuk menggabungkan data-data ujian untuk setiap pembolehubah input yang terlibat dalam setiap soalan aturcara. Namun, masih terdapat kaedah lain yang boleh digunakan untuk membentuk kombinasi data ujian yang diperlukan dalam kes ujian.

Kes ujian yang telah direka bentuk dari data ujian yang telah dijana akan mewakili aktiviti-aktiviti yang akan dilaksanakan untuk menguji aturcara. Pada peringkat ini hanya ralat aturcara dapat dikenal pasti, tetapi tahap kebetulsalahan aturcara belum dapat ditentukan. Bagi membolehkan ukuran tahap kebetulsalahan aturcara diketahui, suatu sukatan perlu diberikan terhadap setiap aktiviti yang terlibat dalam proses pengujian aturcara. Sub topik seterusnya akan membincangkan sukatan yang akan digunakan oleh kajian ini dalam menentukan kebetulsalahan aturcara.

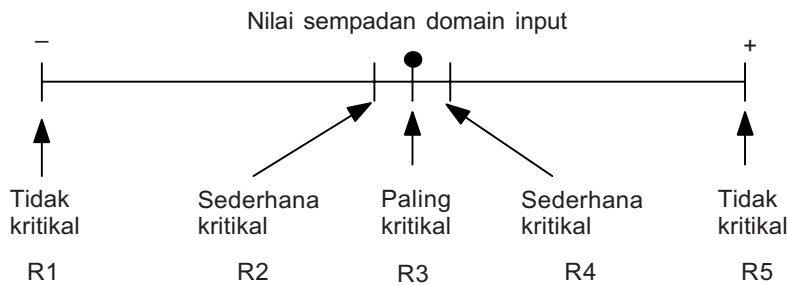
5.0 SKEMA PENJANAAN PEMBERAT UJIAN

Pengujian dinamik melibatkan proses pelaksanaan aturcara terhadap data ujian. Output yang dihasilkan akan dibandingkan dengan output jangkaan bagi menentukan sama ada ianya memenuhi spesifikasi yang telah ditetapkan.

Sukatan kebetulsalahan dinamik bagi kajian ini ditentukan dengan memberikan nilai pemberat bagi setiap data ujian yang dijanakan. Nilai pemberat diberikan berdasarkan paras tahap kritikal data ujian yang dijanakan. Bagi data ujian yang bersifat kritikal, iaitu kecenderungan untuk penghasilan ralat adalah tertinggi, maka nilai pemberat yang tertinggi akan diberikan. Pemberian nilai pemberat untuk setiap data ujian adalah merujuk kepada skema penjanaan data ujian yang telah dijelaskan sebelum ini.

Sebagaimana yang telah dinyatakan, data ujian yang berada hampir dengan sempadan domain input mempunyai kejituan yang tinggi berbanding dengan data ujian yang berada jauh dari sempadan tersebut [15]. Maka data ujian yang mempunyai kejituan yang tinggi dianggap sebagai data ujian yang paling kritikal. Dengan demikian anggapan berikut boleh dibuat: semakin data ujian yang dipilih berjauhan dari sempadan domain input, maka kejituan data ujian tersebut semakin berkurangan. Maka, bolehlah dikatakan bahawa semakin berkurangan kejituan data ujian yang dipilih, semakin kurang kritikal kes ujian berkenaan.

Tahap kritikal setiap kes ujian bagi pembolehubah input numerik bernilai positif atau negatif ditunjukkan oleh Rajah 9.



Rajah 9 Tahap kritikal setiap kes ujian bagi pembolehubah input numerik bernilai positif atau negatif

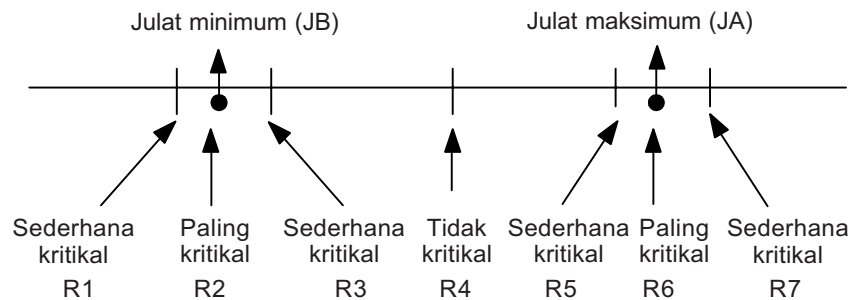
Berdasarkan Rajah 9, skema pemberian nilai pemberat bagi setiap kes ujian untuk pembolehubah input numerik bernilai positif atau negatif ditunjukkan dalam Jadual 4.

Jadual 4 Skema pemberian nilai pemberat bagi kes ujian yang dijana dari pembolehubah input numerik bernilai positif atau negatif

Sempadan kes ujian	Kes ujian	Tahap kritikal kes ujian	Nilai pemberat
Pada sempadan domain input	R3	Paling kritikal	5
Bersebelahan sempadan domain input dengan $+\delta$	R4	Sederhana kritikal	4
Bersebelahan nilai sempadan domain input dengan $-\delta$	R2	Sederhana kritikal	4
Bersebelahan nilai sempadan domain dengan $+\infty$	R5	Tidak kritikal	3
Bersebelahan nilai sempadan domain dengan $-\infty$	R1	Tidak kritikal	3

Merujuk kepada Jadual 4, nilai pemberat diberi berdasarkan tahap kritikal. Nilai pemberat paling tinggi perlu diberikan kepada kes ujian paling kritikal (R3), diikuti dengan pemberat kedua tertinggi untuk sederhana kritikal (R4 dan R2) dan yang paling sedikit pemberatnya untuk tahap tidak kritikal (R5 dan R1). Nilai 5 telah dipilih bagi nilai pemberat paling maksimum untuk contoh ini. Nilai ini dianggap sesuai untuk bilangan data ujian yang dijana bagi setiap pembolehubah input bagi kajian ini. Namun begitu, pemilihan nilai yang lain tidak memberikan kesan perbezaan dari segi markah akhir yang diperolehi dari pengujian dinamik yang dilaksanakan, selagi syarat iaitu tahap kritikal lebih tinggi diberikan nilai pemberat tertinggi dipatuhi. Bagi setiap perbezaan tahap kritikal kes ujian, contoh ini memilih untuk menolak nilai 1 daripada nilai pemberat maksimum.

Tahap kritikal setiap kes ujian bagi pembolehubah input numerik bernilai global ditunjukkan oleh Rajah 10.



Rajah 10 Tahap kritikal bagi setiap kes ujian untuk pembolehubah input numerik bernilai global

Berdasarkan Rajah 10, skema pemberian nilai pemberat dan markah bagi setiap kes ujian untuk pembolehubah input numerik bernilai global ditunjukkan dalam Jadual 5.

Jadual 5 Skema pemberian nilai pemberat dan markah bagi kes ujian yang dijana daripada pembolehubah input numerik yang bernilai global

Sempadan kes ujian	Kes ujian	Tahap kritikal kes ujian	Nilai pemberat
Pada sempadan julat minimum (JB)	R2	Paling kritikal	5
Bersebelahan sempadan julat minimum (JB) dengan $+\delta$	R3	Sederhana kritikal	4
Bersebelahan sempadan julat minimum (JB) dengan $-\delta$	R1	Sederhana kritikal	4
Pada sempadan julat maksimum (JA)	R6	Paling kritikal	5
Bersebelahan sempadan julat maksimum (JA) dengan $+\delta$	R7	Sederhana kritikal	4
Bersebelahan sempadan julat maksimum (JA) dengan $-\delta$	R5	Sederhana kritikal	4
Bersebelahan sempadan julat minimum (JB) dengan $+\infty$ dan bersebelahan sempadan julat maksimum (JA) dengan $-\infty$	R4	Tidak kritikal	3

Dengan merujuk kepada Jadual 5, cara pemberian nilai pemberat yang diberikan untuk setiap kes ujian adalah sama seperti yang dijelaskan dalam Jadual 4. Penyediaan nilai pemberat untuk kedua-dua skema di atas (Jadual 3 dan Jadual 4) dilakukan secara automatik.

5.1 Contoh Pengiraan Markah

Dengan mengambil contoh daripada hasil kes ujian dalam soalan 2, maka pemberat ujian bagi setiap kes ujian adalah seperti berikut:

Kes ujian	Partisi input	Data ujian	Output	Komen	Nilai pemberat
R1	Sempadan bawah partisi (i)	-10	0	Ralat, input tidak sah	3 (Tidak kritikal)
R2	Sempadan atas partisi (i)	-1	0	Ralat, input tidak sah	4 (Sederhana kritikal)
R3	Di luar sempadan atas partisi (i) dan sempadan bawah partisi (ii)	0	0	Input sah	5 (Paling kritikal)
R4	Sempadan bawah partisi (ii)	1	1	Input sah	4 (Sederhana kritikal)
R5	Sempadan atas partisi (ii)	10	10	Input sah	3 (Tidak kritikal)

Ini bermakna, 5 kes ujian dapat disediakan untuk menguji aturcara pelajar bagi soalan 2. Kes ujian R1 dan R2 sepatutnya menghasilkan mesej ralat iaitu "Ralat input tidak sah". Manakala bagi kes ujian R3, R4 dan R5 sepatutnya mengembalikan suatu nilai iaitu 0, 1 dan 10 masing-masing.

Andaikan hanya kes ujian R4 dan R5 yang dipenuhi oleh aturcara pelajar. Maka markah yang diperolehinya dikira seperti berikut (Nilai 0 diberikan jika gagal, dan 1 untuk berjaya):

$$\begin{aligned}
 &= \frac{(0 \times 3) + (0 \times 4) + (0 \times 5) + (1 \times 4) + (1 \times 3)}{3 + 4 + 5 + 4 + 3} \\
 &= 100 \times \frac{7}{19} \\
 &= 36.8\%
 \end{aligned}$$

6.0 KESIMPULAN

Kertas ini membincangkan penggunaan teknik kaedah Nilai Sempadan dalam penjanaan data secara automasi serta memanjangkan perbincangan dengan

mengesyorkan penggunaan teknik ini dalam menentukan pemberat bagi data ujian. Skema untuk penjanaaan data ujian dan pemberat ujian yang telah dibincangkan ini telah diimplementasi bagi penjanaaan data ujian dan sukatan penaksiran kebetulsalahan aturcara bagi Sistem Penaksiran Aturcara C, STaksirC.

RUJUKAN

- [1] Chu, H. D., J. E. Dobson, dan I. C. Liu. 1997. FAST-A Framework for Automating Statistic-based Testing. (dalam talian) <http://citeseer.nj.nec.com/73306.html>. (20 Nov 2001).
- [2] Singh, H., M. Conrad, dan S. Sadeghiporr. 1997. Test Case Design Based on Z and Classification-tree Method. (dalam talian) <http://citeseer.nj.nec.com/singh97test.html> (17 Mei 2002).
- [3] Offutt, A. J., J. Clark, T. Zhang, dan K. Tewary. 1994. Experiments with Data Flow and Mutation Testing. (dalam talian) <http://citeseer.nj.nec.com/offutt94experiments.html>. (22 Nov 2001).
- [4] Zarina, S. 1999. The Automatic Assessment of Z Specification. Tesis Dr. Fal. University of Nottingham.
- [5] Sommerville, I. 1996. *Software Engineering (ed. Ke-5)*. England: Addison-Wesley Publishing Company.
- [6] Al-Amayreh, A. A. R. 1997. PROBE: A Formal Specification-based Testing System. Tesis Sarjana Sains. Universiti Kebangsaan Malaysia.
- [7] Huang, J. C. 2001. Specification-based Test Selection. (dalam talian) <http://www.cs.un.edu/~jhuang/JCH/PJ/3specbased.pdf>. (20 Nov 2001).
- [8] Hameed, S. A. 2000. Meaningful Test Data Generation Based on Statistical Method. Tesis Dr. Fal. University Kebangsaan Malaysia.
- [9] Korel, B. 1996. Automated Test Data Generation for Programs with Procedures. In Proceedings of the 1996 International Symposium on Software Testing and Analysis. ACM Press. 209-215.
- [10] Miller, W., dan D. L. Spooner. 1976. Automatic Generation of Floating Point Test Data. *IEEE TSE*. SE-2(3): 223-226.
- [11] Gallagher, M. J., dan V. L. Narasimhan. 1997. Adtest: A Test Data Generation Suite for Ada System. *IEEE TSE*. 23(8): 473-484.
- [12] Roper, M., I. Maclean, A. Brooks, J. Miller, dan M. Wood. 1995. Genetic Algorithms and Automatic Generation of Test Data. Laporan Teknikal RR/95/195 [Efo CS 19-95]. University of Strathclyde.
- [13] McGraw, G., C. Micheal, dan M. Schatz. 1998. Generating Software Test by Evolution. Technical Report RSTR-018-97-01. (dalam talian) <http://citeseer.nj.nec.com/mcgraw97generating.html> (20 Nov 2001).
- [14] Rohaida, R. 2002. Penjanaaan Data Ujian untuk Penaksiran Automatik Tugasan Aturcara C. Tesis Sarjana. Universiti Kebangsaan Malaysia.
- [15] Tracey, N., J. Clark, K. Mander, dan J. McDermid. 1998. An Automated Framework for Structural Test-Data Generation. (dalam talian) <http://citeseer.nj.nec.com/tracey98automated.html>. (20 Nov 2001).
- [16] IPL Information Processing. 1996. Design Unit Test Cases. (dalam talian). <http://www.iplbath.com/pdf/p0829.pdf>. (20 Mei 2001).
- [17] Mott, L. J., A. Kandel, dan T. P. Baker. 1986. *Discrete Mathematics for Computer Scientists and Mathematicians*. Ed. ke-2. Englewood Cliffs: Prentice-Hall.