

Category Partition Method untuk Program dengan Non-Numerik Input

Arlinta Christy Barus

Del Polytechnic of Informatics

Technical Report

arlinta@del.ac.id

Abstrak

Pada *paper* ini dituliskan hasil perbandingan empiris yang dilakukan pada *Category partition method* mrnggunakan teknik perhitungan jarak 0 dan 1 dan teknik perhitungan jarak 0,1,2,..N-1 pada *Random Testing* dan *Adaptive Random Testing* (ART), dimana N adalah jumlah maksimum partisi dari setiap kategori. *Category partition method* merupakan salah satu *method* dari pendekatan partisi pada *Adaptive Random Testing* (ART). Perbandingan ini diukur dengan menggunakan *F-measure* dan waktu yang dibutuhkan untuk menemukan *failure*. Pada dokumen ini dijelaskan eksperimen yang dilakukan dan hasil dari setiap eksperimen.

Kata kunci: *Category partition method, software testing, Adaptive Random Testing(ART), Random Testing(RT)*.

PENDAHULUAN

Software testing adalah aspek yang penting saat pengembangan *software* untuk meyakinkan kualitas *software* yang dibangun. *Random Testing(RT)* adalah strategi *software testing* yang paling sederhana. RT mengurangi biaya dengan design *test case* dan algoritma yang sederhana saat men-*generate test case*[1].

Dari penelitian yang dilakukan oleh Kwok Ping Chan dkk memperkenalkan sebuah metode baru yang merupakan perkembangan dari metode RT, yaitu *Adaptive Random Testing(ART)*. ART mendistribusikan *test case* lebih merata pada *input domain* dan mendeteksi *failure* dengan *test case* yang lebih sedikit dibandingkan RT (*F-Measure* yang lebih kecil)[3]. Beberapa pendekatan pada ART adalah berdasarkan jarak (*distance*), batas (*exclusion*) dan partisi (*partitioning*).

LATAR BELAKANG

Saat ini banyak *software* yang menerima *inputan numeric* maupun *non numeric*. Namun penelitian yang dilakukan selama ini cenderung kepada *inputan numeric*. Karena secara global pengimplementasian RRT dengan *inputan numeric* akan lebih mudah dibandingkan dengan *inputan non numeric*.

Pada *inputan numeric*, jarak antara *inputan* dapat diketahui dengan menghitung jarak nilai antara dua *inputan*. Tetapi pada RRT dengan *non-numeric input*, konsep seperti yang dipakai untuk mencari jarak antara dua *input* numerik tidak dapat digunakan, sehingga, pada RRT dengan *input non-numerik* dipakai *category partition method* untuk menghitung jarak antara dua *inputan*.

PERMASALAHAN

Pada penelitian dengan non-numerik *input* yang dilakukan sebelumnya, konsep *Category partition* yang digunakan adalah membedakan setiap *inputan* dengan nilai 0 dan 1 saja. Pada paper ini diusulkan konsep *category partition method* yang berbeda dengan yang sudah ada, pada *category partition method* yang sudah ada sekarang hanya membedakan *partition* antara dua *input* dengan nilai 0 jika *partition*-nya sama dan 1 jika *partition*-nya berbeda, sedangkan pada proyek ini akan dibedakan jarak tiap *partition*.

Misalnya setiap *category inputan* pada aplikasi memiliki 3 *partition*, pada konsep *category partition method* yang ada perbedaan antara *partition* pertama ke kategori kedua dan *partition* pertama ke kategori ketiga adalah sama yaitu jarak partisi diberi nilai 1. Pada konsep *category partition method* yang diusulkan pada

proyek ini akan dibedakan nilai jarak antara *partition* satu ke *partition* dua dan *partition* satu ke *partition* tiga. Karena tentu lebih berbeda jarak antara *partition* satu ke *partition* tiga dibandingkan *partition* satu ke *partition* dua.

CATEGORY PARTITION METHOD

Category partition method merupakan salah satu *method* pada RRT. Pada penelitian sebelumnya, metode ini telah dilakukan untuk *inputan numeric* dan *non numeric*. Pada paper ini yang diteliti adakah *Category partition* dengan *inputan non numeric* yang juga sudah pernah diteliti.

Pada penelitian sebelumnya untuk menghitung jarak antar partisi hanya menggunakan jarak 0 jika partisi sama dan 1 jika partisi berbeda, pada paper ini diusulkan perhitungan jarak antar partisi dengan menggunakan nilai 0,1,2,...N-1, dimana N meruakan jumlah partisi pada satu kategori.

Ada 5 langkah menghitung jarak pada RRT dengan *input non numeric*, yaitu:

1. Menentukan kategori dari *input* yang disediakan.
2. Menentukan partisi dari setiap kategori yang telah ditentukan sebelumnya.
3. Memberikan nilai perbandingan antara partisi dari setiap kategori.
4. Masukkan *input* yang ada dan menentukan nilai perbandingan antara *input* yang telah disediakan.
5. Membandingkan hasil dari setiap *input*, kemudian hasil yang sama diberikan nilai 0 dan hasil

yang berbeda diberikan nilai 1,2,3,..N-1, kemudian menjumlahkan hasil dari perbedaan tiap *input*-an.

METRIK

Untuk mengukur keefektifan dari kedua metode, penulis menggunakan ukuran *F-measure*, semakin kecil *F-measure* dari metode yang diusulkan maka metode yang diusulkan lebih baik daripada metode yang sebelumnya. Selain *F-measure*, penulis juga membandingkan waktu yang dibutuhkan masing-masing metode untuk menemukan *failure*, sama halnya dengan *F-measure*, semakin sedikit waktu yang digunakan untuk menemukan *failure* maka metode tersebut akan semakin baik.

EMPIRICAL STUDIES

Object Testing yang digunakan pada eksperimen ini adalah program penerimaan TNI yang terdiri dari 10 kategori, program ini dipilih supaya menggambarkan metode yang diusulkan, jarak antar partisi akan kelihatan bedanya. Dari 10 kategori yang ada, 5 diantaranya terdiri dari 3 partisi sehingga, program ini mendukung eksperimen yang dilakukan terhadap teknik yang dipilih.

Jumlah *test cases* yang menjadi salah satu bahan dalam *software testing* adalah 6.000 *test cases* yang digenerate sendiri oleh penulis dan *test case* yang digenerate memenuhi kriteria lulus, tidak lulus dan dipertimbangkan.

HASIL EKSPERIMEN

Untuk melakukan eksperimen, penulis membangun beberapa *tools* yang dibutuhkan untuk setiap eksperimen, yaitu:

1. *Test case generator*, program ini digunakan untuk men-generate *test case* secara *Random* sebanyak 6.000 *test cases*.

Test case di-generate berdasarkan 10 kategori yang telah disebutkan pada bab2. Setiap kategori di-generate untuk memenuhi kondisi lulus, dipertimbangkan dan tidak lulus. Setiap *test case* yang di-generate terdapat kemungkinan beberapa kategori tidak memiliki nilai atau kosong.

2. *Test frame converter*, program ini digunakan untuk mengkonversi *test case* yang diperoleh menjadi *test frame*. *Test case* yang di-generate di-convert untuk mengetahui partisi dari setiap *test case*.

3. Program simulasi RT, program ini digunakan untuk melakukan simulasi teknik RT terhadap program Penerimaan TNI.

4. Program simulasi RRT A. Dengan teknik perhitungan jarak 0 dan 1. Program ini digunakan untuk melakukan simulasi teknik RRT dengan teknik perhitungan jarak 0 dan 1, bernilai 0 apabila berada pada partisi yang sama dan bernilai 1 jika berada pada partisi yang berbeda.

5. Program simulasi RRT B dengan teknik perhitungan jarak 0,1,2,..N-1. Dimana N merupakan jumlah partisi pada kategori. Program ini digunakan untuk melakukan simulasi terhadap teknik RRT yang diusulkan dengan perhitungan distance 0, 1, 2, N-1 terhadap program Penerimaan TNI.

Program simulasi RT, RRT A dan RRT B terdiri dari program original dan 7 program *mutant*.

KESIMPULAN

Hasil percobaan yang dilakukan pada program RT, RRT A dan RRT B disajikan lebih sistematis pada Tabel 1.

Table 1 Hasil percobaan program RT, RRT A dan RRTB

Dari setiap *mutant*, diambil 100 *F-measure* untuk dianalisis menggunakan SPSS. Pada percobaan yang dilakukan, setiap *mutant* dari setiap *Category* terdiri dari 1000 *F-measure*, sehingga data yang dianalisis sangat banyak. Dengan menggunakan SPSS, setiap *F-measure* dibandingkan untuk mengetahui perbedaan setiap *F-measure* dari setiap metode. Pada percobaan yang dilakukan, *F-measure* metode RRT yang diusulkan lebih kecil daripada RT dan RRT yang sudah ada sekarang, untuk melihat hasil *F-measure* setiap metode menggunakan Duncen dapat dilihat pada Tabel 2 Untuk melihat perbedaan setiap *F-measure* signifikan atau tidak dapat dilihat pada Tabel 3.

Table 2 perbedaan F-measure dengan teknik pengelompokan duncan

Algoritma	N	Subset for alpha = .05	
		1	2
RRT B	700	11.84	
RRT A	700	12.22	
RT	700		20.12
Sig.		.816	1.000

Table 3 perbedaan F-measure dengan menggunakan Anova

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	30592.230	2	15296.115	16.558	.000
Within Groups	1937229.513	2097	923.810		
Total	1967821.742	2099			

Referensi

Mutan	F-measure			Waktu		
	RRT B	RRT A	RT	RRT B	RRT A	RT
1	37.042	31.769	63.006	4001.0 m	2797.0 m	44.0 m
2	210.2	2.362	3.487	166.0 m	88.0 m	4.0 m
3	40.664	34.174	56.162	4111.0 m	2623.0 m	41.0 m
4	3.484	3.253	5.710	363.0 m	123.0 m	6.0 m
5	1.882	1.788	1.738	146.0 m	55.0 m	1.0m
6	6.605	4.352	8.364	677.0 m	160.0 m	8.0 m
7	2.076	2.357	3.282	166.0 m	87.0 m	4.0 m

- [1] K.P. Chan, T. Y. Chen, and D. Towey, "Restricted Random Testing", Department of Computer Science and Information Systems, *University of Hong Kong*, Hong Kong
- [2] T.J. Ostrand, M.J. Balcer, The category-partition method for specifying and generating functional test, *Communications of the ACM* 31 (6) (1988) 676–686, June 1988
- [3] Diana, "Test Case Distributions Of Adaptive Random Testing", pp40-55.