# MODUL 3 PENGUJIAN DOMAIN *(DOMAIN TESTING)*

|  |  |
| --- | --- |
|  | CAPAIANPEMBELAJARAN |
| 1. Mahasiswa memahami mekanisme pengujian domain 2. Mahasiswa memahami dasar-dasar pengujian domain 3. Mahasiswa dapat menguji kesalahan domain 4. Mahasiswa mampu membuat *test case* untuk pengujian domain 5. Mahasiswa mampu menganalisis hasil pengujian domain | |

|  |  |
| --- | --- |
|  | KEBUTUHAN ALAT/BAHAN/SOFTWARE |
| 1. Alat yang dibutuhkan : Seperangkat komputer 2. Perangkat Lunak yang dibutuhkan : Microsoft Office Visio /Dia, Java, Notepad | |
|  | DASAR TEORI |

Dua elemen dasar dari suatu program computer adalah domain masukan dan alur program.Domain masukan dari suatu program adalah kumpulan dari semua data masukan pada program.Alur program adalah urutan perintah dari awal program hingga sejumlah titik perhatian dalam program.Sebagai contoh, akhir program adalah titik perhatian. Titik perhatian lainnya adalah ketika program menunggu menerima masukan lain dari lingkungannya, sehingga bisa dilanjutkan eksekusinya. Dengan kata lain, alur program, atau alur sederhana, menghubungkan sejumlah aliran kendali dalam program. Suatu alur disebut layak jika disana terdapat data masukan yang menyebabkan program mengeksekusi alur.

Terdapat dua kelas kesalahan, yaitu kesalahan komputasi (*computation error*) dan kesalahan domain (*domain error*), dengan mengkombinasikan konsep data masukan dan alur program.

*Computation error*.Suatu kesalahan komputasi terjadi bila data masukan tertentu menyebabkan program mengeksekusi dengan benar misalnya alur yang dikehendali, tetapi nilai keluarnya salah.Catatan bahwa nilai keluaran dapat salah bahkan jika alur yang dikehendaki telah dieksekusi.Hal ini dapat terjadi karena fungsi yang seharusnya salah dieksekusi dalam pernyataan tugas. Misalnya, alur yang dikehendaki berisi hasil pernyataan = f(a,b), dimana a dan b nilai masukan. Kesalahan komputasi dapat terjadi jika pernyataan diganti dengan satu kesalahan, seperti hasil = f(b,a). meskipun, hasil mengeksekusi alur bisa salah karena dari suatu kesalahan didalam pernyataan tugas, dan ini dapat terjadi saat mengeksekusi alur yang benar.

*Domain error* suatu kesalahan domain terjadi bial data masukan tertentu menyebabkan program salah mengeksekusi, pada alur program yang tidak dikehendaki.Alur yang salah dapat dipilih oleh suatu program jika disana salah pada satu atau lebih pernyataan kondisional dalam program. Misal pernyataan kondisional dari bentuk if(p)then f1() else f2(). Jika ada suatu kesalahan dalam perumusan dari predikat p, kemudian fungsi pemanggil salah memanggil, maka menyebabkan eksekusi pada alur yang salah.

Kedua jenis kesalahan program untuk meunjukkan program computer untuk melaksanakan pemetakan abstraksi fungsi. Idealnya, setiap nilai masukan, alur program yang sama dapat dieksekusi secara eksklusif dengan subset nilai input. Disini setiap subset nilai input, menyebabkan alur yang sama dieksekusi berdasarkan pada subdomain atau domain masukan. Sehingga, dikatakan program digunakan untuk memetakan domain kedalam alurnya sendiri. Gambar-3.1 menjelaskan tentang konsep domain program. Kelompok D adalah kelompok semua input dari program P (gambar 3.1.a). D adalah domain dari seluruh program. kelompok D dibatasi dengan kelompoknya, dan P tidak memiliki perilaku komputasi yang berbeda untuk setiap elemen D. Misal, P dapat melakukan komputasi yang sama untuk semua elemen subset tertentu dari D. Misal, seperti yang ditunjukkan di gambar 3.1.b, P melakukan 5 komputasi yang berbeda, satu untuk setiap subset D1, dua untuk subset D2, dst. Ini dapat dikatakan bahwa pemecahan D tidak mungkin berada diluar P. Misal, P mempunyai mekanisme seperti yang digambarkan pada gambar 3.1c, untuk memutuskan metode perhitungan yang diperlukan dengan memilih cabag tertentu bila P memanggil input tertentu. Pengklasifikasi input tidak mungkin ada dalam satu program tunggal, bentuk pengenal yang jelas. Konsep dapat berada dalam entitas yang bisa ditemukan sebagai konsep cross-cutting; cross cutting karena bagian-bagian dari pengklasifikasi input ditemukan pada modul program yang berbeda. Pada gambar 3.1c memperlihatkan 5 perhitungn berbeda, perhitungan untuk D1 melalui perhitugan untuk D1, dst. Selanjutnya, suatu program akan melakukan perhitungan yang salah jika ada kesalahan (fault) dalam pengklasifikasian bagian input. Berikut ini beberapa definisi dari istilah diatas.



Gambar 3.1 Konsep domain program

* **Domain** adalah kelompok dari nilai input yang mana program melakukan perhitungan yang sama pada setiap anggota dari kelompok. Kita berminat dalam domain maksimal yang mana bahwa program melakukan perhitungan berbeda pada domain tertentu.
* **Program** dikatakan memiliki kesalahan domain (*domain error*) jika program melakukan kesalahan dalam mengklasifikasikan input. Diasumsikan domain tertentu melakukan perhitungan berbeda, kesalahan doamn akan menyebabkan program menghasilkan ouput yang salah.

Menguji Kesalahan Domain

Ide pengujian domain pertama dilakukan oleh White dan Cohen pada tahun 1978. Ada perbedaan mendasar antara teknik pengujian berbasis flow grap dan pengujian domain. Dengan flow graph kita berarti membuat CFG dan DFG. Perbedaannya dijelaskan sebagai berikut:

* Memilih alur dari CFG atau DFG sesuai dengan kriteria soverage tertentu. Hal yang sama, kriteria yang ditinjau untuk menemukan definisi dan menggunakan dua aspek variabel dalam suatu program merupakan all-defs, all-c-uses, dan all-uses untuk memilih alur dan menghubungkan dengan data uji, tidak ada asumsi yang dibuat terkait jenis actual dari fault pada kasusnuji yang dipilih yang berpotensi tidak ditemukan, disana tidak ada jenis khusu dari fault secara eksplisit dapat dideteksi.
* Menetapkan domain pengujian pada seluruh pendekatan baru untuk mendeteksi fault. Pertama mendefinisikan kategori dari fault, yang disebut domain error, dan memilih data uji untuk mendeteksi fault tersebut. Jika program memiliki domain error, yang akan dinyatakan dengan kasus uji.

Konsep lebih detail adalah sebagai berikut:

* *Sources of Domain*: berarti suatu program contoh, kita menjelaskan bagaimana predikat program mempunyai klasifikasi input.
* *Type of Domain Error*: Menjelaskan bagimana modifikasi kecil dilakukan pada predikat program, yang dapat diinterpretasikan sebagai cacat program, yang dinyatakan domain error.
* *Selecting Test Daa to Reveal Error*: uji pemilihan kriteria menjelaskan pengambilan nilai input. Data uji hingga memilih jenis tertentu dari domain error

***Source of Domain***

Domain dapat diidentifikasi dari program maupun spesifikasinya. Metode untuk mengidentifikasi domain dari kode sumber menggunakan langka-langkah berikut:

* Menggambar CFG dari kode sumber
* Mencari semua kemungkinan interpretasi dari predikat. Dengan kata lain, mengungkapkan sendiri predikat dalam hal ini vektor input dan, mungkin vektor dari konstanta. Pembaca bisa mencatat bahwa predikat dalam suatu program dapat mempunyai banyak interpretasi, karena kendali dapat sampai node predikat melalui alur yang berbeda.
* Menganalisis predikat yang diinterpretasikan untuk mengidentifikasi domain.

|  |  |
| --- | --- |
|  | PRAKTIK |

Berikut, akan dijelaskan tentang prosedur untuk mengidentifikasi domain. Diketahui suatu fungsi bahasa C pada Gambar-3.2 yang menggambarkan suatu prosedur untuk mengidentifikasi domain.

Fungsi menerima dua input X dan Y, dan nilai pengembalian (*return*) integer. Gambaran CFG dari *codedomain()* ditunjukkan pada gambar 3.3. Dua predikat dalam dua perintah if() yang ditunjukkan dengan node 3 dan 6 pada gambar 3.3.

Predikat:

P1 : c > 5

int codedomain(int x, int y){

int c, d, k

c = x + y;

if (c *>*5) d = c - x/2;

else d = c + x/2;

if (d *>*= c + 2) k = x + d/2;

else k = y + d/4;

return(k);

}

Gambar-3.2 Fungsi untuk menjelaskan domain program.

**Interpretasi Kode Program**

Pada perintah if() pertama hanya mempunyai satu interpretasi, yaitu:

P1 : x + y > 5

Karena kendali program mencapai perintah if() hanya melalui satu *path* dari *node* awal.



Gambar-3.3 CFG Representasi dari Fungsi Gambar 3.2

P2: d > c + 2

Pada perintah if() kedua terdapat dua interpretasi, karena kendali program dapat mencapai perintah if() kedua melalui 2 *path*: (a) bila evaluasi if() pertama adalah *true* dan (b) bila evaluasi if() pertama adalah *false*. Dua interpretasi ini diringkas dalam tabel 3.1.

Tabel-3.1 Dua Interpretasi dari pernyataan if() kedua gambar-3.2

|  |  |
| --- | --- |
| **Evaluasi dari**  **P1** | **Interpretasi dari**  **P2** |
| True | X < - 4 |
| False | X > 4 |

Dijelaskan prosedur untuk memperoleh domain dari interpretasi P1 dan P2 (Gambar-3.3).

**Analisis Domain Program**

Terdapat grid dua dimensi yang dinyatakan dengan X dan Y pada gambar 3.4.Ukuran grid terlalu besar untuk menampilkan seluruh domain dari program (gambar-3.3).Predikat P1 dibagi kedalam 2 wilayah. Batas P1 ditunjukkan dengan garis lurus yang dinyatakan dengan persamaan x + y = 5.

Selanjutnya, terdapat dua interpretasi dari predikat P2, yaitu untuk P1 = *True*, maka P2 mempunyai interpretasi berikut:

P2 : x < - 4

Oleh karena itu, selanjutnya P2 dibagi kedalam area (kumpulan dari titik-titik), yang didefinisikan dengan P1 = *True* kedalam dua kumpulan yang sesuai dengan dua nilai kebenaran. Untuk batas P2, bila P1 dievaluasi dengan *True*, dinyatakan dengan garis lurus x = -4. Pada area sisi kiri batas P2 dan diatas batas P1 yang sesuai dengan P1P2 = TT, dan area dikanan batas P2 dan diatas batas P1 sesuai dengan P1P2 = TF.

Untuk P1 = False, maka P2 mempunyai interpretasi berikut ini:

P2: x > 4

Dengan kata lain, P2 dibagi kedalam area (kumpulan titik-titik), yang didefinisikan dengan P1 = *False* kedalam dua kumpulan yang sesuai dengan dua nilai kebenaran. Batas P2, bila P1 dievaluasi dengan *False*, dinyatakan dengan garis lurus x = 4. Maka daerah disisi kanan batas dari P2 dan dibawah batas sesuai dengan P1P2 = FT, dan area disisi kiri dari P2 dan batas P1 sesuai dengan P1P2 = FF dalam gambar-3.4.



**Jenis dari Domain Error**

* Domain adalah sekumpulan dari nilai pada program yang melakukan perhitungan identic
* Domain dapat dinyatakan dengan sekumpulan dari predikat. Setiap elemen dari domain sesuai dengan predikat dari domain.

Contoh: domain TT dalam gambar-3.4 dinyatakan secara matematis dengan kumpulan predikat yang ditunjukkan gambar-3.5.

Domain didefinisikan, dari suatu pandangan geometri, dengan sekumpulan dari kendala yang disebut ketidaksamaan batas.Property dari domain dijelaskan sebagai berikut.



|  |  |
| --- | --- |
|  | **LATIHAN** |

Berikut ini diberikan kode program binary search. Input array V[] diasumsikan disimpan urut naik, n adalah ukuran array, dan anda akan mencari indeks dari elemen X dalam array. Jika X tidak ada dalam array, rutin akan memberikan nilai kembalian (return -1).

int binsearch(int X, int V[], int n)

{

int low, high, mid;

low = 0;

high = n - 1;

while (low <= high) {

mid = (low + high)/2;

if (X < V[mid])

high = mid - 1;

else if (X > V[mid])

low = mid + 1;

else

return mid;

}

return -1;

}

Dari kode program fungsi binSearch tersebut selesaikan berikut ini.

1. Buatlah CFG dari binSearch?
2. Dari CFG tersebut, tentukan jalur entry dan exit yang memenuhi kriteria cakupan tersebut?
3. Buatlah jalur independen (independent paths) dari CFG tersebut?
4. Ujilah menggunakan kasus uji, sehingga seluruh jalur terlewati?
5. Buatlah deskripsi pengujian dari kasus tersebut?

|  |  |
| --- | --- |
| **tugas.png** | **TUGAS** |

Buatlah program konversi suhu dari Celcius ke Fahrenheit dan Reamur, Kelvin kemudian buatlah CFG dan cek menggunakan 5 buah data uji , serta buatlah analisisnya.

Dimana nilai konversinya adalah sebagai berikut :

F = C + 237

R = 0,8 x C

K = 1,8 x C

|  |  |
| --- | --- |
|  | **REFERENSI** |

1. Naik, Kshrisagar, Priyadarshi Tripathy, 2008, Software Testing and Quality Assurance Theory and Practice, John Wiley & Sons, Inc., New Jersey, USA.
2. Galin, Daniel, 2006, Software Quality Assurance: From Theory to Implementation, Person, England.