

White Box Testing

Merupakan metode perancangan test case yang menggunakan struktur kontrol dari perancangan prosedural untuk mendapatkan test case.

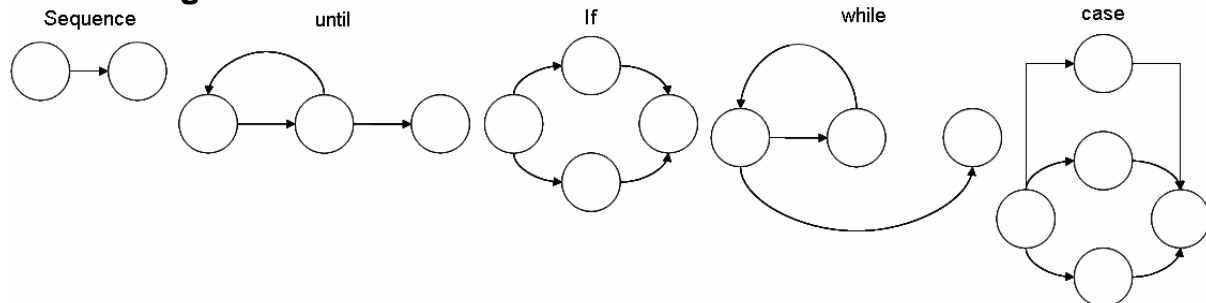
Dengan menggunakan metode white box, analisis sistem akan dapat memperoleh test case yang:

- Menjamin seluruh independent path di dalam modul yang dikerjakan sekurang-kurangnya sekali
- Mengerjakan seluruh keputusan logikal
- Mengerjakan seluruh loop yang sesuai dengan batasannya
- Mengerjakan seluruh struktur data internal yang menjamin validitas

A. Uji Coba Basis Path

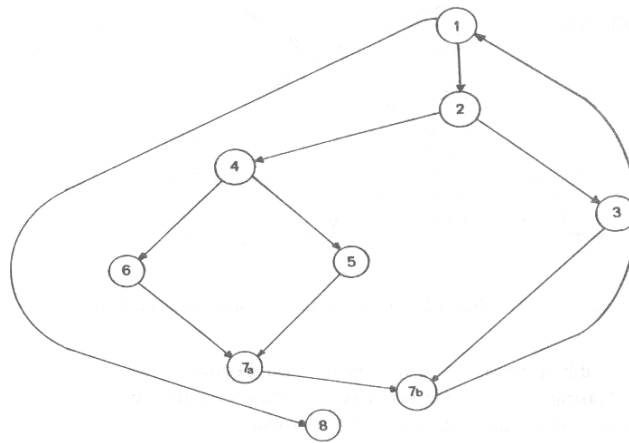
Merupakan teknik uji coba white box yang diusulkan Tom McCabe. Metode ini memungkinkan perancang test case mendapatkan ukuran kekompleksan logical dari perancangan prosedural dan menggunakan ukuran ini sebagai petunjuk untuk mendefinisikan basis set dari jalur pengerjaan. Test case yang didapat digunakan untuk mengerjakan basis set yang menjamin pengerjaan setiap perintah minimal satu kali selama uji coba.

Notasi Diagram Alir

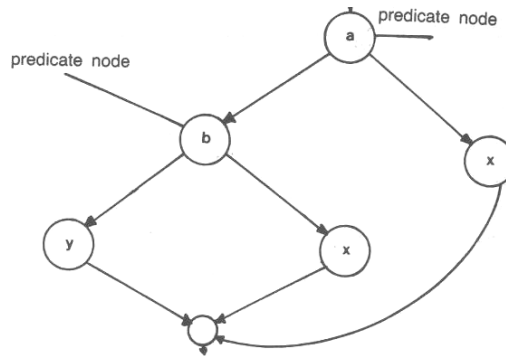


Lingkaran (node), menggambarkan satu/lebih perintah prosedural. Urutan proses dan keputusan dapat dipetakan dalam satu node. *Tanda panah (edge)*, menggambarkan aliran kontrol. Setiap node harus mempunyai tujuan node. *Region* adalah daerah yang dibatasi oleh edge dan node. Termasuk daerah diluar grafik alir. Contoh menterjemahkan pseudo code ke grafik alir

```
1 : do while record masih ada
    baca record
2 : if record ke 1 = 0
3 : then proses record
    simpan di buffer
    naikan counter
4 : else if record ke 2 = 0
5 :     then riser counter
6 :     proses record
    simpan pada file
7a: endi f
    endi f
7b: enddo
8 : end
```



Nomor pada pseudo code berhubungan dengan nomor node. Apabila ditemukan kondisi majemuk (compound condition) pada pseudo code pembuatan grafik alir menjadi rumit. Kondisi majemuk mungkin terjadi pada operator Boolean (AND, OR, NAND, NOR) yang dipakai pada perintah if. Contoh :



```

if A or B
  then procedure x
  else procedure y
endif

```

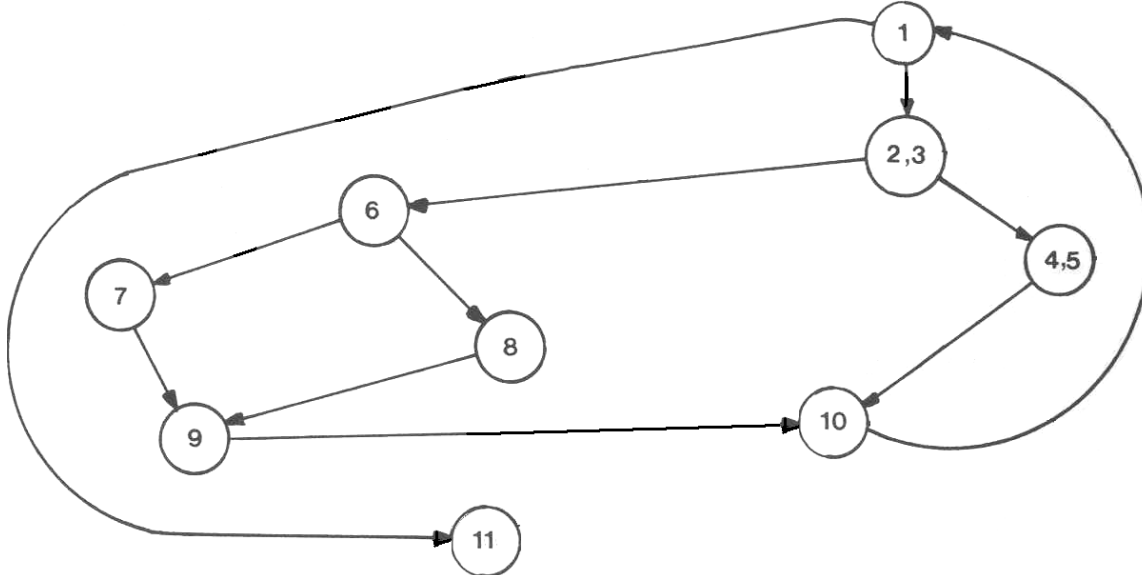
Gambar Logika Gabungan

Node dibuat terpisah untuk masing-masing kondisi A dan B dari pernyataan IF A OR B. Masing-masing node berisi kondisi yang disebut **pridicate node** dan mempunyai karakteristik dua atau lebih edge darinya.

Cyclomatic Complexity

Cyclomatic complexity adalah metrik software yang menyediakan ukuran kuantitatif dari kekompleksan logikal program. Apabila digunakan dalam konteks metode uji coba basis path, nilai yang dihitung untuk cyclomatic complexity menentukan jumlah jalur independen dalam basis set suatu program dan memberi batas atas untuk jumlah uji coba yang harus dikerjakan untuk menjamin bahwa seluruh perintah sekurang-kurangnya telah dikerjakan sekali.

Jalur independent adalah jalur yang melintasi atau melalui program dimana sekurang-kurangnya terdapat proses perintah yang baru atau kondisi yang baru.



Dari gambar:

Path 1 = 1 - 11

Path 2 = 1 - 2 - 3 - 4 - 5 - 10 - 1 - 11

Path 3 = 1 - 2 - 3 - 6 - 8 - 9 ...: 10 - 1 - 11

Path 4 = 1 - 2 - 3 - 6 - 7 - 9 - 10 - 1 - 11

Path 1,2,3,4 yang telah didefinisikan diatas merupakan basis set untuk diagram alir.

Cyclomatic complexity digunakan untuk mencari jumlah path dalam satu flowgraph. Dapat dipergunakan rumusan sebagai berikut :

1. Jumlah region grafik alir sesuai dengan cyclomatic complexity.
2. Cyclomatic complexity $V(G)$ untuk grafik alir dihitung dengan rumus:

$$V(G) = E - N + 2$$

Dimana:

E = jumlah edge pada grafik alir

N = jumlah node pada grafik alir

3. Cyclomatic complexity $V(G)$ juga dapat dihitung dengan rumus:

$$V(G) = P + 1$$

Dimana P = jumlah predicate node pada grafik alir

Pada Gambar dapat dihitung cyclomatic complexity:

1. Flowgraph mempunyai 4 region
2. $V(G) = 11 \text{ edge} - 9 \text{ node} + 2 = 4$
3. $V(G) = 3 \text{ predicate node} + 1 = 4$

Jadi cyclomatic complexity untuk flowgraph adalah 4

Melakukan Test Case

Metode uji coba basis path juga dapat diterapkan pada perancangan prosedural rinci atau program sumber. Prosedur rata-rata pada bagian berikut akan digunakan sebagai contoh dalam pembuatan test case.

```
PROCEDURE RATA-RATA
INTERFACE RESULT rata, total, input, total.valid
INTERFACE RESULT nilai, minim, max
TYPE NILAI (1:100) IS SCALAR ARRAY;
TYPE rata, total, input, total.valid, max, minim, jumlah IS SCALAR;
TYPE I IS INTEGER;
I = 1;
total.input = total.valid = 0;
jumlah = 0;
DO WHILE nilai(i) <> -999 .and. total.input < 100
    tambahkan total.input dengan 1;
    IF nilai(i) >= minimum .and. nilai(i) <=max;
        THEN tambahkan total.valid dengan 1;
        jumlah=jumlah + nilai(i);
    ELSE skip;
    END IF
    tambahkan i dengan 1;
ENDDO
IF total.valid > 0
    THEN rata =jumlah/total.valid;
ELSE rata = -999;
ENDIF
END
```

Langkah-langkah pembuatan test case:

1. Dengan mempergunakan perancangan prosedural atau program sumber sebagai dasar, digambarkan diagram alirnya.

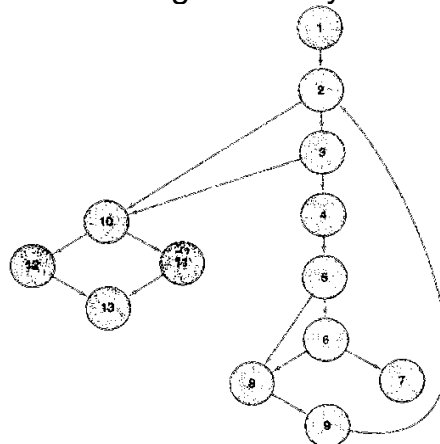


Diagram Alir prosedur rata

2. Tentukan cyclomatic complexity untuk diagram alir yang telah dibuat:

$$V(G) = 6 \text{ region}$$

$$V(G) = 17 \text{ edge} - 13 \text{ node} + 2 = 6$$

$$V(G) = 5 \text{ predicate node} + 1 = 6$$

3. Tentukan independent path pada flowgraph

Dari hasil perhitungan cyclomatic complexity terdapat 6 independent path yaitu:

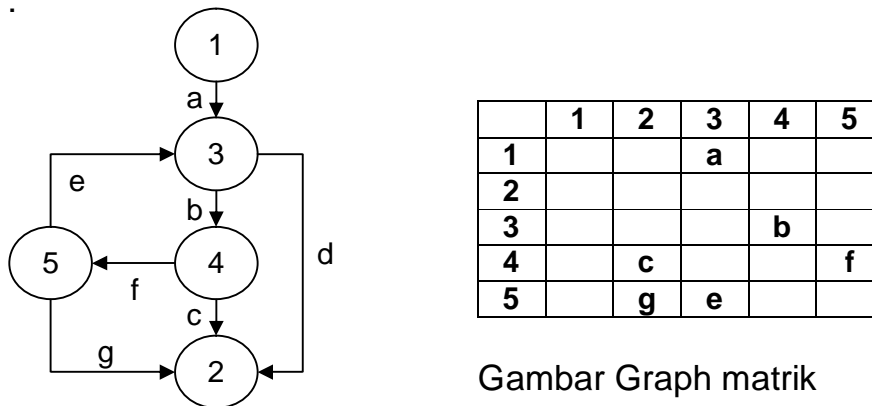
- path 1 : 1-2-10-11-13
- path 2 : 1-2-10-12-13
- path 3 : 1-2-3-10-11-13
- path 4 : 1-2-3-4-5-8-9-2-..
- path 5 : 1-2-3-4-5-6-8-9-2-..
- path 6 : 1-2-3-4-5-6-7-8-9-2-...

4. Buat test case yang akan mengerjakan masing-masing path pada basis set. Data yang dipilih harus tepat sehingga setiap kondisi dari predicate node dikerjakan semua.

Graph Metrik

Graph metrik merupakan software yang dikembangkan untuk membantu uji coba basis path atau struktur data. Graph metrik adalah matrik empat persegi yang mempunyai ukuran yang sama dengan jumlah node pada flowgraph. Masing-masing baris dan kolom mempunyai hubungan dengan node yang telah ditentukan dan pemasukan data matrik berhubungan dengan hubungan (edge) antar node.

Contoh sederhana pemakaian graph metrik dapat digambarkan sebagai berikut :



Gambar Graph matrik

Hubungan bobot menyediakan tambahan informasi tentang aliran kontrol. Secara simpel hubungan bobot dapat diberi nilai 1 jika ada hubungan antara node atau nilai 0 jika tidak ada hubungan. Dapat juga hubungan bobot diberi tanda dengan:

- Kemungkinan link (edge) dikerjakan
- Waktu yang digunakan untuk proses selama traversal dari link
- Memori yang diperlukan selama traversal link
- Sumber daya yang diperlukan selama traversal link

B. Pengujian Loop

Loop merupakan kendala yang sering muncul untuk menerapkan algoritma dengan tepat. Uji coba loop merupakan teknik pengujian white box yang fokusnya pada validitas dari loop. Kelas loop yaitu : loop sederhana, loop tersarang, loop terangkai, loop tidak terstruktur