White Box Testing

#2

- Testing & Implementasi
  Pertemuan 4

- Elkin Rilvani
  elkinrilvani@gmail.com

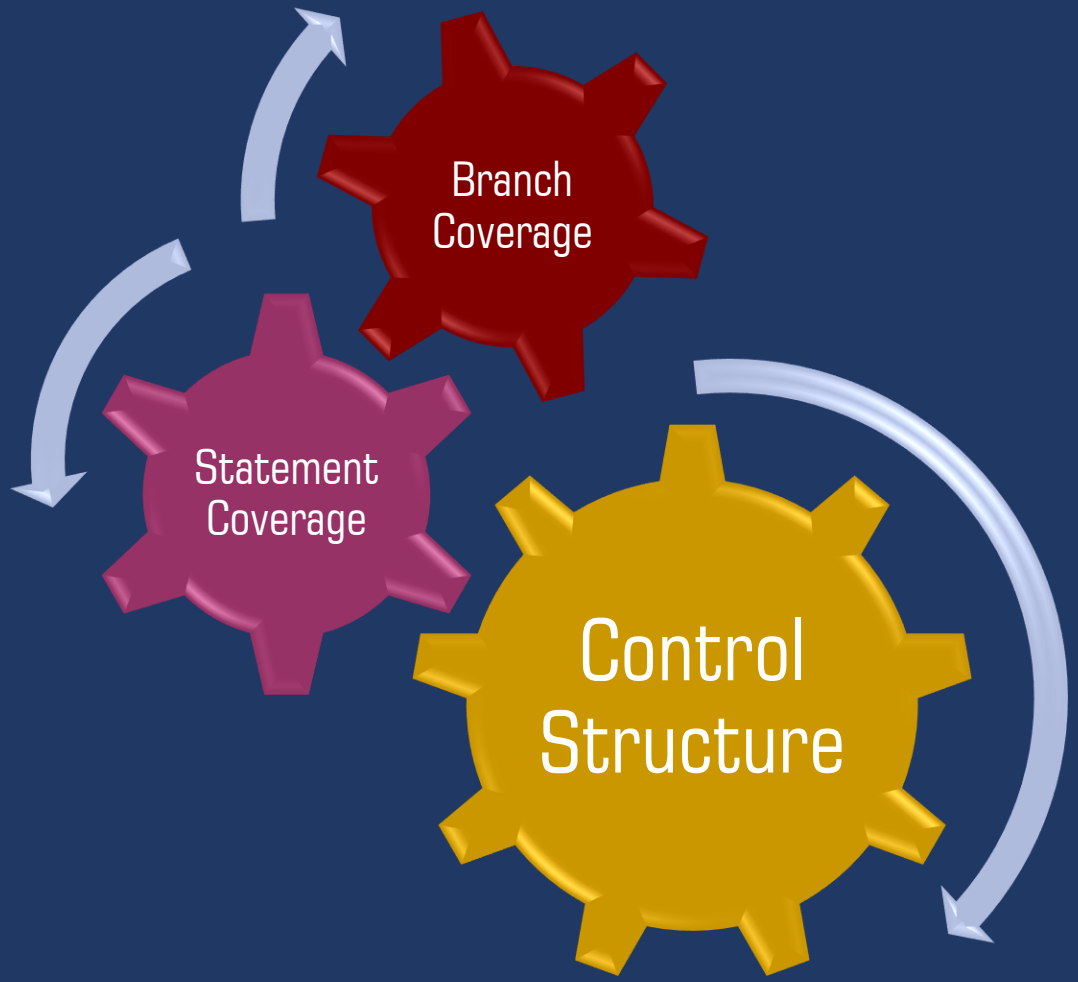
Control Structure

- Condition Testing
- Data Flow Testing
- Loop Testing

Control Structure = Control Flow Testing

- It is a testing technique that comes under white box testing
- Seluruh struktur, desain, dan kode perangkat lunak harus dipelajari untuk jenis pengujian ini.
- Seringkali metode pengujian digunakan oleh pengembang sendiri untuk menguji kode dan desain mereka sendiri karena mereka sangat terbiasa dengan kode tersebut.
- Metode ini diimplementasikan dengan maksud untuk menguji logika kode sehingga hasil yang dikehendaki atau fungsi bisa tercapai.

Now, we will define various coverage methods:

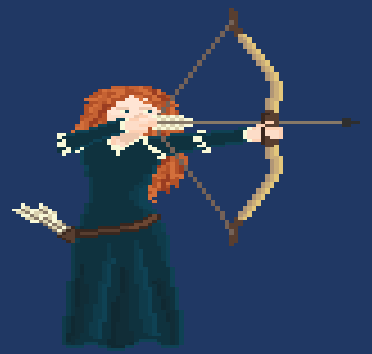


Statement Coverage

- Segala sesuatu yang kurang dari 100% Statement Coverage berarti tidak semua baris kode telah dieksekusi.
- Untuk dapat menyelesaikan Statement Coverage dengan mengidentifikasi nomor **Cyclomatic** yang mengeksekusi sekali rangkaian kasus uji ini.
- Keuntungan dari Statement Coverage adalah sangat mudah memisahkan bagian dari kode, yang tidak dapat dijalankan

Karena cenderung menjadi mahal, pengembang memilih teknik pengujian yang lebih baik

Branch Coverage

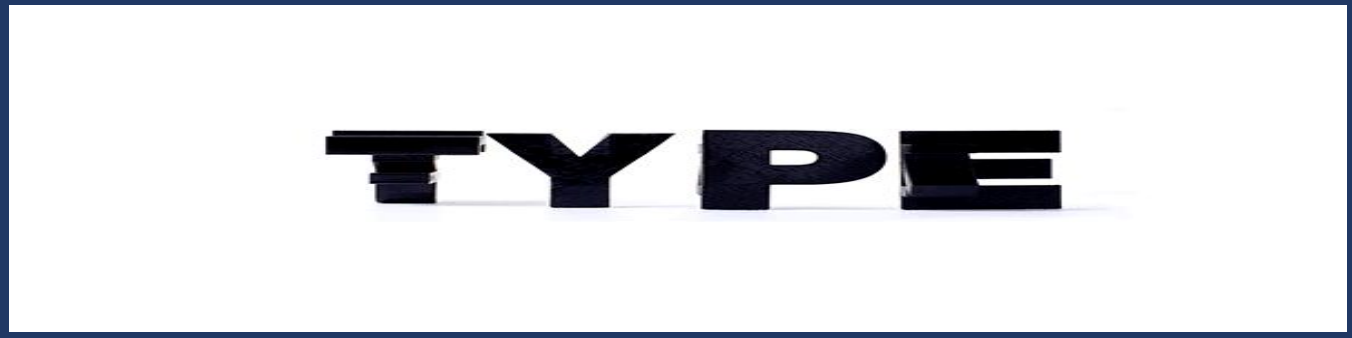


- Coverage Kriteria pengujian yang lebih baik adalah Branch Coverage or Decision coverage.
- Ukuran persentase titik keputusan program yang telah dievaluasi baik yang benar maupun yang salah dalam kasus uji
- Examples of branch coverage-DO statements, IF statements and multiway GOTO statements.
- Branch coverage biasanya untuk menunjukkan Statement Coverage yakin berjalan 100% setiap grafik aliran kontrol dilalui.

Condition Coverage

Condition Testing

- Coverage Kriteria yang lebih kuat pengujian adalah Condition testing
- Pengujian mengukuran persentase sub-ekspresi Boolean dari program yang telah dievaluasi sebagai hasil benar dan salah dalam kasus uji.
- Operator boolean yang dapat digunakan dalam suatu kondisi komplek adalah OR (|), AND (&) dan NOT (–).



Error pada Condition Testing adalah sebagai berikut:

- Kesalahan operator Boolean
- Kesalahan variabel Boolean
- Kesalahan boolean parentheses
- Kesalahan operator relasional
- Kesalahan ekspresi aritmatika

ADVANTAGE

- Pengukuran Condition testing yang dites adalah sederhana
- Condition testing program yang dites menyediakan tuntunan untuk pembuatan tes tambahan bagi program.

Data Flow Testing

- Which looks at how data moves within a program.
- In data flow testing the control flow graph, dianotasikan dengan informasi tentang bagaimana variabel program didefinisikan dan digunakan.
- Bisa definisikan data flow testing sebagai teknik pengujian yang didasarkan pada pengamatan bahwa nilai yang terkait dengan variabel dapat mempengaruhi eksekusi program
- Data flow testing memilih jalur yang cukup untuk memastikan bahwa:

1. Setiap objek data telah diinisialisasi sebelum penggunaannya.
2. Semua objek yang didefinisikan telah digunakan setidaknya satu kali.

Some of the Important points of data flow testing are:

1. Semua anomali arus data terpecahkan.
2. Hindari masalah integrasi dengan melakukan semua data flow pada suatu variabel dalam rutinitas yang sama.
3. Bila memungkinkan gunakan deklarasi eksplisit data.

Data flow testing cenderung menguak bug seperti variabel yang digunakan namun tidak diinisialisasi atau dideklarasikan namun tidak digunakan, seterusnya.

```
int
sum(int *A, int n)
{
  int x, i = 0;

  while(i < n) {
    x = x + A[i];
    ++i;
  }
  return x;
}
```

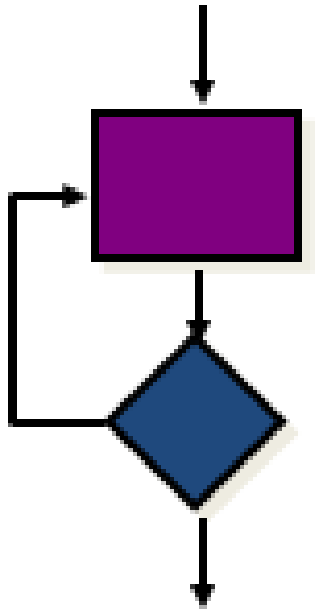
x is not initialized

Loop Testing

- Testing which exclusively focuses on the validity of loop construct.
- Loop sederhana untuk diuji kecuali ada dependensi antara loop atau di antara loop dan kode yang dikandungnya.
- There are four classes of loops:

1. Simple Loop
2. Nested Loop
3. Concatenated Loop
4. Unstructured Loop

Simple Loop



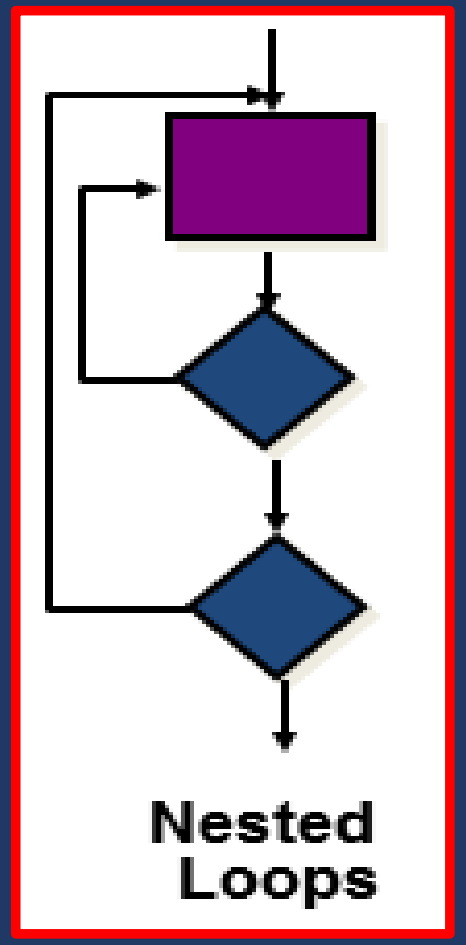
Simple loop

```
public class DoWhileExample {
public static void main(String[] args) {
    int i=1;
    do{
        System.out.println(i);
    i++;
    }while(i<=10);
}
}
```

Simple Loop

- Set tes berikut dapat diterapkan pada loop sederhana, di mana n itu adalah no maksimum. lolos melewati loop. [13]
- Step 1: Lewati loop seluruhnya
- Step 2: Hanya satu yang melewati loop
- Step 3: Dua melewati loop
- Step 4: m melewati loop dimana n> m
- Step 5: n-1, n, n + 1 melewati loop

Nested Loop




```java
/**
 * Print a 10 x 10 times table.
 * @parm args the command-line arguments
 */
public static void main(String[] args)
{
    out.println();
    out.println("A Times Table");
    out.println("==========================================================");

    for (int outer = 1; outer <= 10; outer++)
    {
        for (int inner = 1; inner <= 10; inner++)
        {
            int product = inner * outer;
            out.printf("%5d", product);
        }
        out.println();
    }
}
```

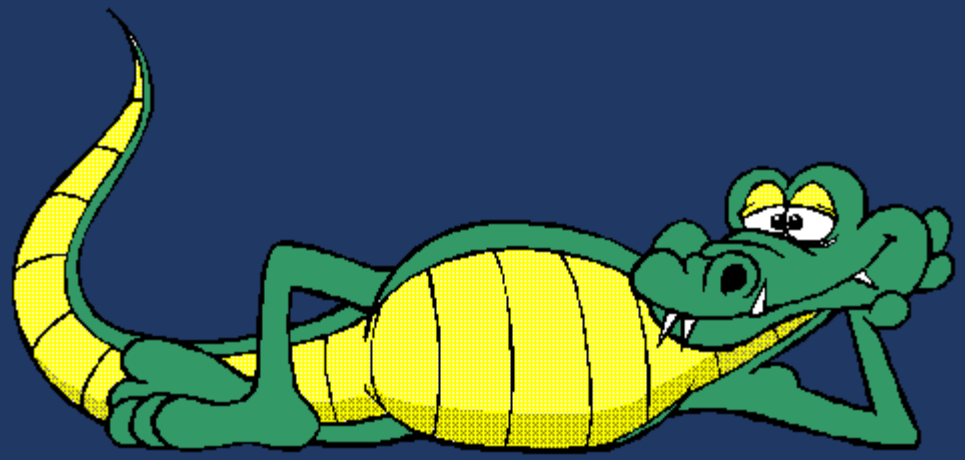
1

2

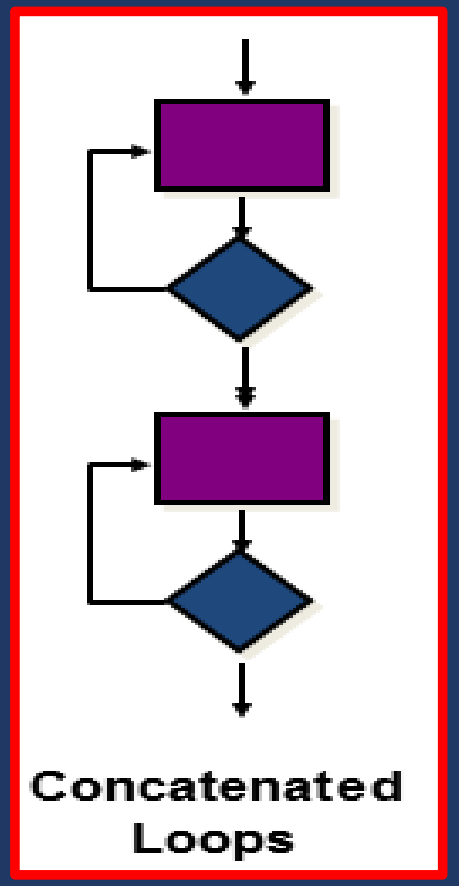
3

Nested Loop

- Langkah pengujian yang dilakukan
- Step 1 : Mulailah dari lingkaran paling dalam
- Step 2 : Melakukan tes untuk loop berikutnya dan bekerja ke luar
- Step 3: Lanjutkan sampai semua loop telah diuji



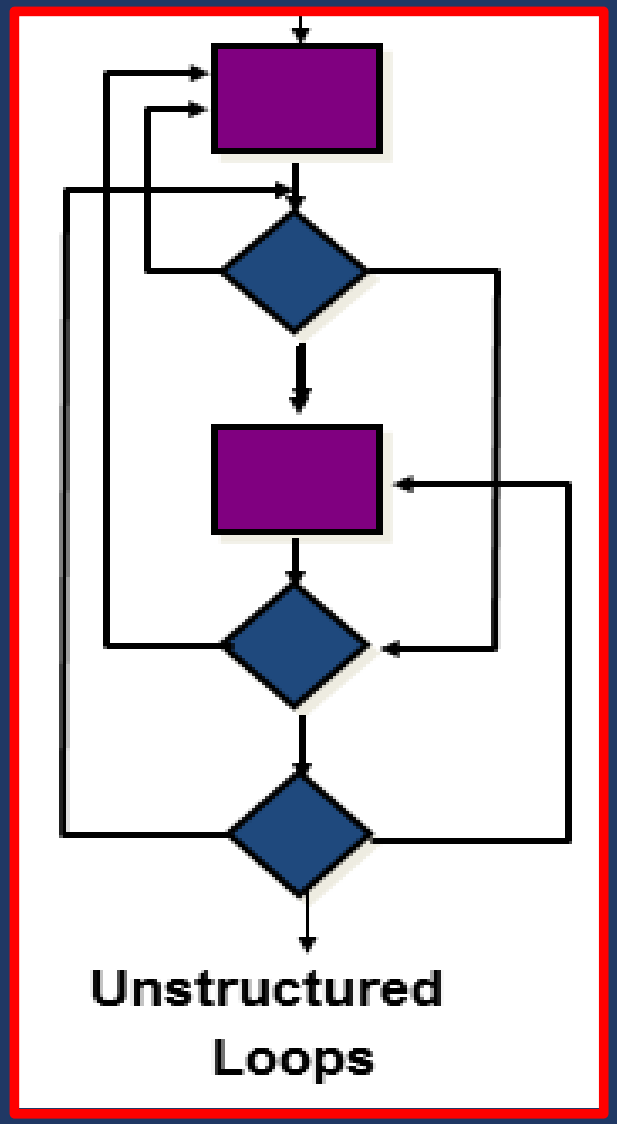
Concatenated Loop




```
public String concatenateStringbuilder() {
    StringBuilder s = new StringBuilder();
    for (int i = 0; i < getCount(); i++) {
        s.append(String.valueOf(i));
    }
    return s.toString();
}

public String concatenatePlus() {
    String s = "";
    for (int i = 0; i < getCount(); i++) {
        s += String.valueOf(i);
    }
    return s.toString();
}
```

Unstructured Loop




```
      PROGRAM CABLE
C
C  THIS PROGRAM COMPUTES THE VELOCITY OF A CABLE CAR ON A THOUSAND-FOOT
C  CABLE WITH THREE TOWERS
C
      INTEGER TOTDIS, DIST, TOWER
C
      WRITE(*,1)
    1 FORMAT('1',9X,'CABLE CAR REPORT'/' ','DISTANCE',
     +       2X,'NEAREST TOWER',2X,'VELOCITY'/1X,2X,
     +       '(FT)',19X,'(FT/SEC)')
C
      TOTDIS = 0
C
    5 IF(TOTDIS.GT.250)GO TO 10
      TOWER = 1
      DIST = TOTDIS
      GO TO 20
C
   10 IF(TOTDIS.GT.750)GO TO 15
      TOWER = 2
      DIST = IABS(TOTDIS - 500)
      GO TO 20
C
   15 TOWER = 3
      DIST = 1000 - TOTDIS
C
   20 IF(DIST.LE.30)VEL = 2.425+0.00175*DIST*DIST
      IF(DIST.GT.30)VEL = 0.625+0.12*DIST-0.00025*DIST*DIST
C
      WRITE(*,40)TOTDIS, TOWER, VEL
   40 FORMAT(' ',I4,11X,I1,9X,F7.2)
C
      TOTDIS=TOTDIS+10
C
      IF(TOTDIS.LE.1000)GO TO 5
C
      STOP
      END
```

```
      program cable
c
c  this program computes the velocity of a cable car on a thousand-foot
c  cable with three towers
c
      integer totdis, dist, tower

      write(*,1)
    1 format('1',9x,'cable car report'/' ','distance',
     +       2x,'nearest tower',2x,'velocity'/1x,2x,
     +       '(ft)',19x,'(ft/sec)')

      totdis = 0

    5 if(totdis.gt.250)go to 10
      tower = 1
      dist = totdis
      go to 20

   10 if(totdis.gt.750)go to 15
      tower = 2
      dist = iabs(totdis - 500)
      go to 20

   15 tower = 3
      dist = 1000 - totdis

   20 if(dist.le.30)vel = 2.425+0.00175*dist*dist
      if(dist.gt.30)vel = 0.625+0.12*dist-0.00025*dist*dist

      write(*,40)totdis, tower, vel
   40 format(' ',i4,11x,i1,9x,f7.2)

      totdis=totdis+10

      if(totdis.le.1000)go to 5

      stop
      end
```

goto constructs an if-else

goto constructs a loop


Thank you ...