Pertemuan 7

Perulangan DO-WHILE, perintah BREAK dan CONTINUE

Mata Kuliah : Algoritma & Pemrograman

Dosen : Tessy Badriyah, SKom., MT., PhD.

Tujuan Pembelajaran

- Mempelajari penggunaan perulangan DO .. WHILE
- Mempelajari penggunaan perintah BREAK dan CONTINUE pada perulangan.

DO ... WHILE LOOP

Mata Kuliah : Algoritma & Pemrograman

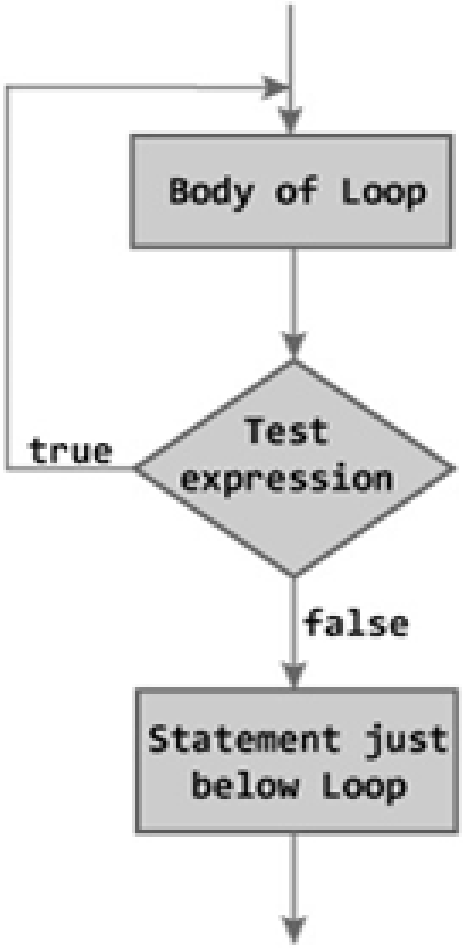
Dosen : Tessy Badriyah, SKom., MT., PhD.

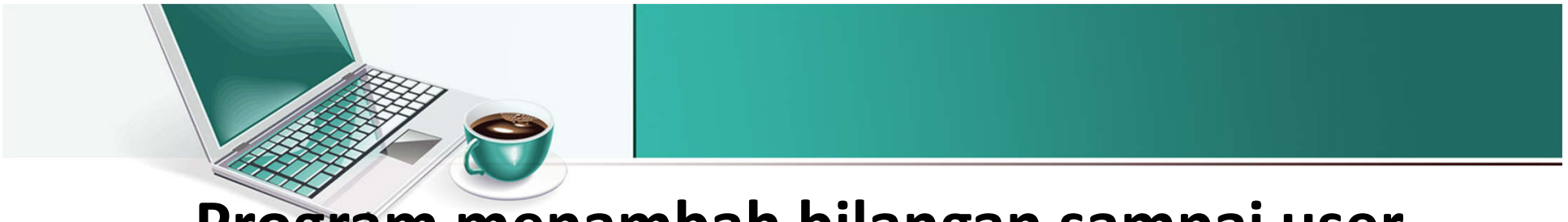
do...while loop

- do..while loop mirip dengan while loop dengan satu perbedaan
- Kalau do...while loop, tubuh perulangan (body of loop) dieksekusi sedikitnya sekali sebelum ujiEkspresi diperiksa nilainya.
- Berikut sintak (cara penulisan) do ... While loop

```
do
{
  // kode
}
while (testExpression);
```

Flowchart dari do...while loop





Program menambah bilangan sampai user memasukkan angka 0

```
#include <stdio.h>
int main()
{   double bilangan, total = 0;
   // tubuh perulangan dijalankan sedikitnya sekali
do
   {
      printf("Masukkan bilangan: ");
      scanf("%lf", &bilangan);
      total += bilangan;
   }
   while(bilangan != 0.0);
   printf("total = %.2lf",total);
   return 0;
}
```

Output:

```
Masukkan bilangan: 1.5
Masukkan bilangan: 2.4
Masukkan bilangan: -3.4
Masukkan bilangan: 4.2
Masukkan bilangan: 0
total = 4.70
```

Deret Fibonacci

- Deret Fibonacci adalah rangkaian dimana bilangan urutan berikutnya adalah jumlah dari dua bilangan sebelumnya.
- Berikut deret Fibonacci :

The Fibonacci sequence: 0, 1, 1, 2, 3, 5, 8, 13, 21

Program deret Fibonacci

```
#include <stdio.h>
int main()
{
   int i, n, t1 = 0, t2 = 1, nextTerm = 0;
   printf("Enter the bilangan of terms: ");
   scanf("%d", &n);
   printf("Fibonacci Series: ");
   for (i = 1; i <= n; ++i)
   {
      // Prints the first two terms.
      if(i == 1)
      {
         printf("%d, ", t1);
         continue;
      }
      if(i == 2)
      {
         printf("%d, ", t2);
         continue;
      }
      nextTerm = t1 + t2;
      t1 = t2;
      t2 = nextTerm;
      printf("%d, ", nextTerm);
   }
   return 0;
}
```

Output:
Masukkan n integer: 10
Factorial of 10 = 3628800

Mencari total bilangan dengan while Loop

```
#include <stdio.h>
int main()
{
   int n, i, total = 0;
   printf("Masukkan  positive integer: ");
   scanf("%d",&n);
   i = 1;
   while ( i <=n )
   {
      total += i;
      ++i;
   }
   printf("total = %d",total);
   return 0;
}
```

Output:

Masukkan positive integer: 100

total = 5050

Program untuk input bilangan positif

```
#include <stdio.h>
int main()
{   int n, i, total = 0;
    do {
        printf("Masukkan  positive integer: ");
        scanf("%d",&n);
    }
    while (n <= 0);
    for(i=1; i <= n; ++i)
    {
        total += i;   // total = total+i;
    }
    printf("total = %d",total);
    return 0;
}
```

BREAK AND CONTINUE

Mata Kuliah : Algoritma & Pemrograman

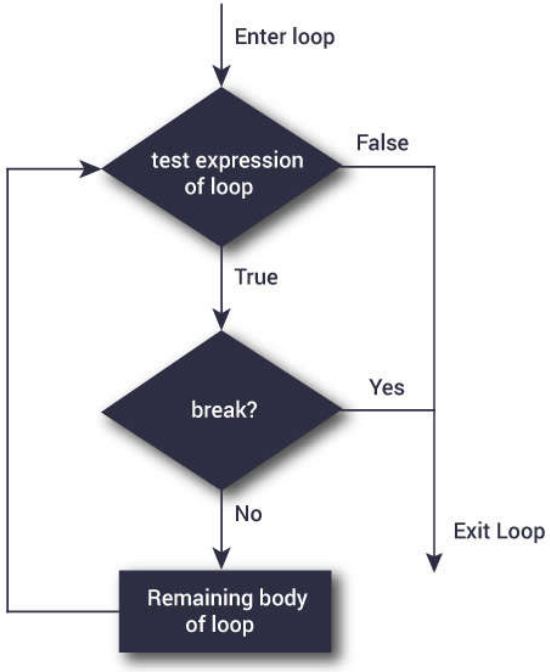
Dosen : Tessy Badriyah, SKom., MT., PhD.

break and continue statement

- It is sometimes desirable to skip some statements inside the loop or terminate the loop immediately without checking the test expression.
- In such cases, break and continue statements are used.
- break and continue statements can alter the program flow of loops.

break Statement

- The break statement terminates the loop (for, while and do...while loop) immediately when it is encountered.
- The break statement is used with decision making statement such as if...else.
- Syntax of break statement : break;
- Flowchart of break statement



Cara kerja perintah break

```
while (test Expression)
{
    // codes
    if (condition for break)
    {
        break;
    }
    // codes
}
```

```
for (init, condition, update)
{
    // codes
    if (condition for break)
    {
        break;
    }
    // codes
}
```

Contoh penggunaan BREAK

```
# include <stdio.h>
int main()
{   int i;
    double bilangan, total = 0.0;
    for(i=1; i <= 10; ++i)
    {
        printf("Masukkan n%d: ",i);
        scanf("%lf",&bilangan);
// Jika yang diinputkan bilangan negatif, maka loop akan dihentikan
        if(bilangan < 0.0)
        {
            break;
        }
        total += bilangan; // total = total + bilangan;
    }
    printf("total = %.2lf",total);
return 0;
}
```

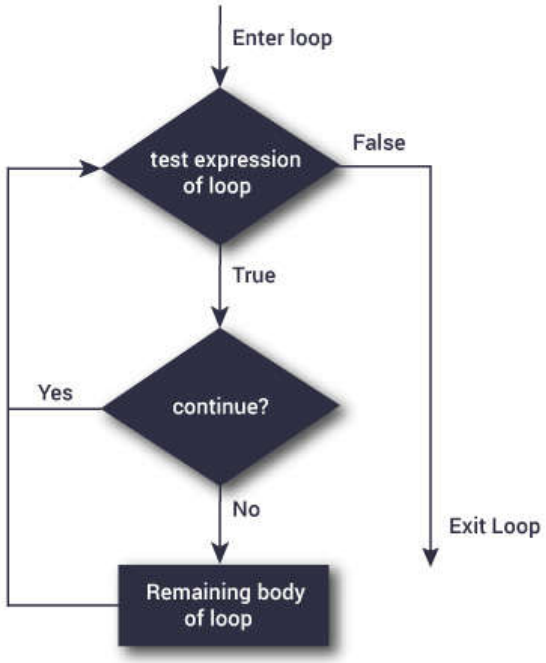
Output :

```
Masukkan  n1: 2.4
Masukkan  n2: 4.5
Masukkan  n3: 3.4
Masukkan  n4: -3
total = 10.30
```

Program akan menghitung jumlah total bilangan, maksimum sampai 10 bilangan. Jika sebelum counter=10, user menginputkan bilangan negatif, maka program akan berhenti

Perintah continue

- Perintah continue akan meloncati (skip) perintah yang ada di dalam loop
- Perintah continue digunakan bersama dengan pernyataan pengambilan keputusan if...else.
- Berikut Flowchart dari perintah continue



Cara kerja perintah continue

```
while (test Expression)
{
    // codes
    if (condition for continue)
    {
        continue;
    }
    // codes
}
```

```
for (init, condition, update)
{
    // codes
    if (condition for continue)
    {
        continue;
    }
    // codes
}
```

Contoh program dengan perintah continue

```
// Program untuk menghitung total maksimum 10 bilangan
// Bilangan negatif yang dimasukkan akan di-skip pada perhitungan
# include <stdio.h>
int main()
{   int i;
    double bilangan, total = 0.0;
    for(i=1; i <= 10; ++i)
    {
        printf("Masukkan  n%d: ",i);
        scanf("%lf",&bilangan);
// Jika user memasukkan bilangan negatif, maka perintah dibawah continue akan di skip
        if(bilangan < 0.0)
        {
            continue;
        }
        total += bilangan; // total = total + bilangan;
    }
    printf("total = %.2lf",total);
    return 0;
}
```

Output :

```
Masukkan  n1: 1.1
Masukkan  n2: 2.2
Masukkan  n3: 5.5
Masukkan  n4: 4.4
Masukkan  n5: -3.4
Masukkan  n6: -45.5
Masukkan  n7: 34.5
Masukkan  n8: -4.2
Masukkan  n9: -1000
Masukkan  n10: 12
total = 59.70
```

Yang sudah dipelajari

- Mempelajari penggunaan perulangan DO .. WHILE
- Mempelajari penggunaan perintah BREAK dan CONTINUE pada perulangan.

Referensi

- Robertson, Lesley Anne. (1992). *Students' guide to program design*. Oxford : Newnes
- Santner, Williams, and Notz (2003), Design and Analysis of Computer Experiments, Springer.
- Deitel & Deitel, *C How to Program*, Prentice Hall 1994 (2nd edition)
- Brookshear, J.G., *Computer Science: An Overview*, Benjamin-Cummings 2000 (6th edition)
- Kernighan & Ritchie, *The C Programming Language*, Prentice Hall