



# **TIB21 – PEMROGRAMAN DASAR**

---

**U N I V E R S I T A S   B U N D A   M U L I A**



# **PENGULANGAN DALAM BAHASA PEMROGRAMAN C++**

**Pertemuan ke-7 dan 8**

# Sub-CPMK

Mahasiswa mampu mengaplikasikan program control bahasa pemrograman C++ untuk kasus perulangan (C3, A3)

## Materi

1. Struktur pengulangan dalam bahasa C++
2. Konstruksi For dalam bahasa C++
3. Konstruksi While dalam bahasa C++
4. Konstruksi Repeat dalam bahasa C++

# Referensi

1. Referensi 1, Bab 4 poin 4.8 (hal 117-118), Bab 5 (hal 159-187)
2. Referensi 2, Bab 8, hal 215-226



# **1. Struktur pengulangan dalam bahasa C++**

- Struktur control pengulangan digunakan untuk melaksanakan satu atau beberapa instruksi secara berulang-ulang.
- Frekuensi pengulangan dapat ditentukan dalam program atau ditentukan saat program dijalankan.
- Meskipun dalam pengulangan ini instruksi yang sama dikerjakan berulang-ulang, namun nilai variable pengendali kondisi pengulangan harus berubah pada setiap iterasi pengulangan agar pengulangan bisa berakhir.

- Terdapat tiga instruksi pengulangan, yaitu: for, while, dan do...while.

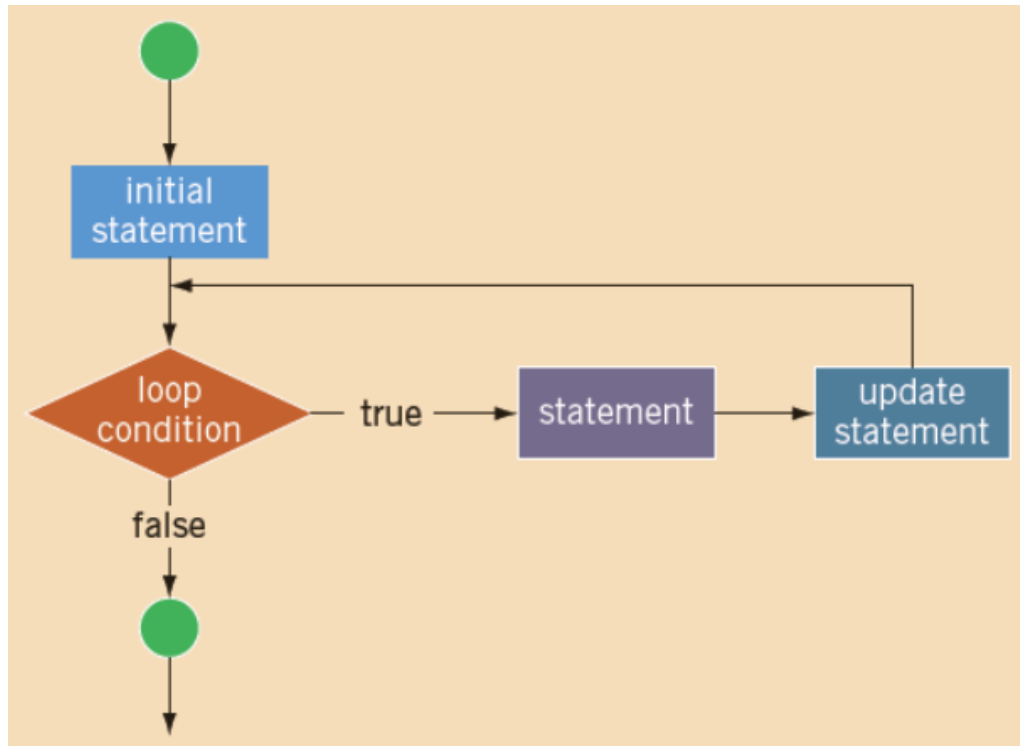


## 2. Konstruksi For dalam bahasa C++



## 2.1 Struktur For

`for(initial statement; loop condition; update statement)  
statement`



## 2.2 Contoh

### 2.2.1 Menampilkan Angka Ascending

```
9  #include <iostream>
10 using namespace std;
11
12 int main()
13 {
14     int i;
15     for (i = 0; i < 10; i++)
16         cout << i << " ";
17     cout << endl;
18 }
19
```



0 1 2 3 4 5 6 7 8 9

Program ended with exit code: 0

## 2.2 Contoh (Lanj..)

### 2.2.2 Menampilkan Angka Descending

```
9  #include <iostream>
10 using namespace std;
11
12 int main()
13 {
14     int i;
15     for (i = 10; i >= 1; i--)
16         cout << i << " ";
17     cout << endl;
18 }
19
```



10 9 8 7 6 5 4 3 2 1

Program ended with exit code: 0

## 2.2 Contoh (Lanj..)

### 2.2.3 Menampilkan Angka Pangkat Dua

```
9  #include <iostream>
10 using namespace std;
11
12 int main()
13 {
14     int i;
15     for (i=2; i<100; i=2*i)
16         cout << i << " ";
17     cout << endl;
18 }
19
```



2 4 8 16 32 64

Program ended with exit code: 0

## 2.2 Contoh (Lanj..)

### 2.2.4 Menampilkan Pattern Bintang

```
9  #include <iostream>
10 using namespace std;
11
12 int main(){
13     int i, j;
14     for (i = 1; i <= 5; i++){
15         for (j = 1; j <= i; j++){
16             cout << "*";
17             cout << endl;
18         }
19     return 0;
20 }
21
```



```
*
**
***
****
*****
```

Program ended with exit code: 0

```
9  #include <iostream>
10 using namespace std;
11
12 int main(){
13     int i, j;
14     for (i = 5; i >= 1; i--){
15         for (j = 1; j <= i; j++){
16             cout << "*";
17             cout << endl;
18         }
19     return 0;
20 }
21
```



```
*****
****
***
**
*
```

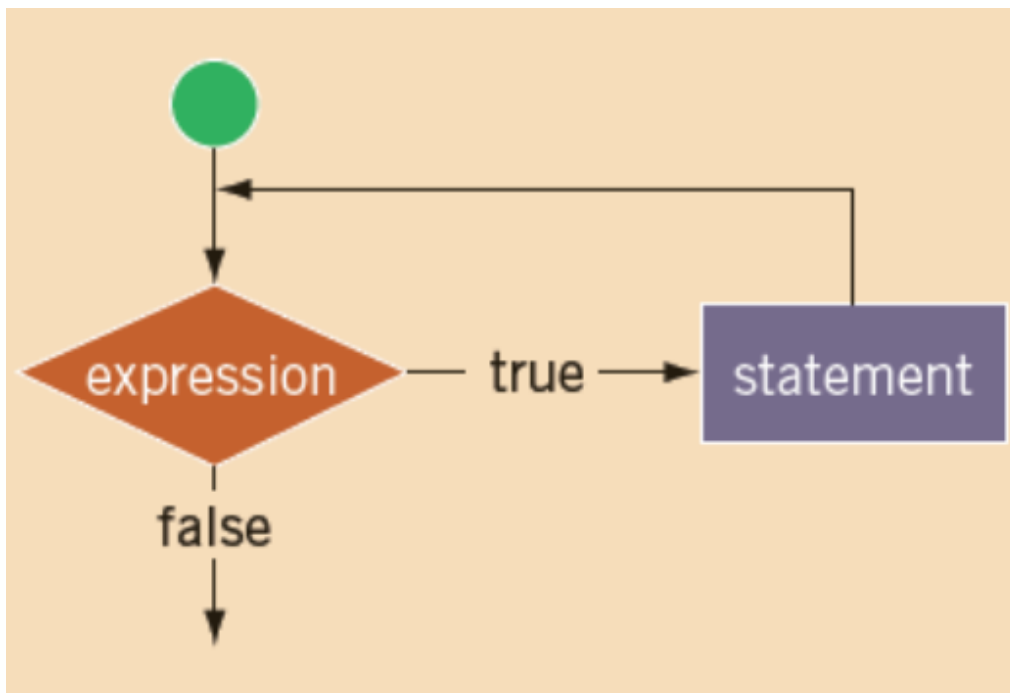
Program ended with exit code: 0



### **3. Konstruksi While dalam bahasa C++**

## 3.1 Struktur While

While (**expression**)  
**statement**



## 3.2 Ekuivalensi **for** dan **while**

**for(initial statement; logical expression; update expression)  
statement**

**Initial expression**

**while(expression)**

**{**

**statement**

**update expression**

**}**



## 3.2 Ekuivalensi **for** dan **while** (Lanj..)

```
for(int i=0; i<10; i++)  
    cout << i << " ";  
cout << endl;
```

---

```
int i=0;  
while(i<10)  
{  
    cout << i << " ";  
    i++;  
}  
cout << endl;
```

## 3.3 Contoh

### 3.3.1 Program Hitung Kalori

```
9  #include <iostream>
10 using namespace std;
11
12 int main()
13 {
14     int calBurnedInADay;
15     int calBurnedInAWeek;
16     int day;
17
18     day = 1;           //inisialisasi awal
19     calBurnedInAWeek = 0; //inisialisasi awal
20
21     while(day<=7)
22     {
23         cout << "Enter clories burned day " << day << ": ";
24         cin >> calBurnedInADay;
25         cout << endl;
26
27         calBurnedInAWeek = calBurnedInAWeek + calBurnedInADay;
28         day = day + 1;
29     }
30
31     cout << "Average number of calories burned each day: "
32          << calBurnedInAWeek / 7 << endl;
33     return 0;
34 }
```

#### Output:

```
Enter clories burned day 1: 300
Enter clories burned day 2: 200
Enter clories burned day 3: 175
Enter clories burned day 4: 270
Enter clories burned day 5: 330
Enter clories burned day 6: 6198
Enter clories burned day 7: 306

Average number of calories burned each day: 254
Program ended with exit code: 0
```

## 3.3 Contoh (Lanj..)

### 3.3.2 Program Menampilkan Bilangan Kelipatan Lima

```
9  //Program menampilkan
10 //bilangan kelipatan lima
11 #include <iostream>
12 using namespace std;
13
14 int main(){
15     int i = 0;
16
17     while(i<=20)
18     {
19         cout << i << " ";
20         i = i + 5;
21     }
22
23     cout << endl;
24 }
```

Output:

```
0 5 10 15 20
Program ended with exit code: 0
```

# 3.3 Contoh (Lanj..)

## 3.3.2 Program Menampilkan Bilangan Kelipatan Lima (Lanj..)

Iteration	Value of i	Expression in Line 2	Statements in Lines 4 and 5
1	i = 0	i <= 20 is true	Output: 0 i = i + 5 = 0 + 5 = 5
2	i = 5	i <= 20 is true	Output: 5 i = i + 5 = 5 + 5 = 10
3	i = 10	i <= 20 is true	Output: 10 i = i + 5 = 10 + 5 = 15
4	i = 15	i <= 20 is true	Output: 15 i = i + 5 = 15 + 5 = 20
5	i = 20	i <= 20 is true	Output: 20 i = i + 5 = 20 + 5 = 25
6	i = 25	i <= 20 is false	The loop terminates

## 3.4 Counter-Controlled **while** Loops

- Digunakan jika kita sudah mengetahui jumlah pengulangan di awal.
- Maka jika **counter** < **n**, aksi yang berada di dalam **while** akan dieksekusi.
- Struktur:

```
counter=0 //inisialisasi variable counter
while (counter<N)
{
    ...
    Counter++;
    ...
}
```

## 3.4 Counter-Controlled **while** Loops (Lanj..)

- Mahasiswa di Universitas Bunda Mulia pada masa pandemic ini banyak yang berjualan cookies untuk menambah uang jajannya. Masing-masing mahasiswa melaporkan jumlah kotak yang terjual.
- Kita akan menuliskan program dengan output jumlah kotak yang terjual, total pendapatan dari hasil penjualan cookies, dan rata-rata kotak yang terjual oleh tiap mahasiswa.

## 3.4 Counter-Controlled while Loops (Lanj..)

```
13 #include <iostream>
14 #include <string>
15 #include <iomanip>
16 using namespace std;
17 int main()
18 {
19     string name;
20     int numOfStudents;
21     int numOfBoxesSold;
22     int totalNumOfBoxesSold;
23     int counter;
24     double costOfOneBox;
25     cout << fixed << showpoint << setprecision(2);
26     cout << "Jumlah mahasiswa yang menjual cookies: ";
27     cin >> numOfStudents;
28     cout << endl;
29
30     totalNumOfBoxesSold = 0;
31     counter = 0;
32     while (counter < numOfStudents)
33     {
34         cout << "Nama penjual"
35              << " dan jumlah kotak yang terjual: ";
36         cin >> name >> numOfBoxesSold;
37         cout << endl;
38         totalNumOfBoxesSold = totalNumOfBoxesSold
39                               + numOfBoxesSold;
40         counter++;
41     }
```

```
42     cout << "Total kotak yang terjual: "
43           << totalNumOfBoxesSold << endl;
44     cout << "Masukkan harga/kotak: Rp. ";
45     cin >> costOfOneBox;
46     cout << endl;
47     cout << "Total pendapatan cookies: Rp. "
48           << totalNumOfBoxesSold * costOfOneBox << endl;
49     if (counter != 0)
50         cout << "Rata-rata jumlah kotak yang berhasil"
51              << " dijual oleh masing-masing mahasiswa: "
52              << totalNumOfBoxesSold / counter << endl;
53     else
54         cout << "No input." << endl;
55     return 0;
56
57 }
```

## 3.4 Counter-Controlled **while** Loops (Lanj..)

Output:

```
Jumlah mahasiswa yang menjual cookies: 5

Nama penjual dan jumlah kotak yang terjual: Roxy 10

Nama penjual dan jumlah kotak yang terjual: Melos 20

Nama penjual dan jumlah kotak yang terjual: Molly 30

Nama penjual dan jumlah kotak yang terjual: Petir 25

Nama penjual dan jumlah kotak yang terjual: Seli 15

Total kotak yang terjual: 100
Masukkan harga/kotak: Rp. 20000

Total pendapatan cookies: Rp. 2000000.00
Rata-rata jumlah kotak yang berhasil dijual oleh masing-masing mahasiswa: 20
Program ended with exit code: 0
```



## 3.5 Sentinel-Controlled **while** Loops

- Sentinel bisa digunakan sebagai tanda untuk menghentikan pengulangan, sehingga program akan berjalan terus-menerus dan hanya akan berhenti ketika program membaca nilai sentinel tertentu.
- Struktur:

```
cin >> variable;  
while(variable != sentinel) {  
    ...  
    cin >> variable;  
    ...  
}
```

## 3.5 Sentinel-Controlled **while** Loops (Lanj...)

- Contoh: berdasarkan program sebelumnya, tetapi dengan asumsi kita tidak mengetahui jumlah mahasiswa yang berjualan cookies, maka kita dapat menggunakan nilai -1 sebagai tanda program berhenti membaca jumlah mahasiswa yang berjualan kue.

## 3.5 Sentinel-Controlled while Loops (Lanj...)

```
9  #include <iostream>
10 #include <string>
11 #include <iomanip>
12 using namespace std;
13 const string SENTINEL = "-1";
14 int main() {
15     string name;
16     int numOfStudents;
17     int numOfBoxesSold;
18     int totalNumOfBoxesSold;
19     double costOfOneBox;
20     cout << fixed << showpoint << setprecision(2);
21     cout << "Tuliskan nama penjual"
22           << " dan jumlah kotak yang terjual. "
23           << "Akhiri dengan -1 : " << endl;
24
25     totalNumOfBoxesSold = 0;
26     numOfStudents = 0;
27
28     cin >> name;
29
30     while (name != SENTINEL) {
31         cin >> numOfBoxesSold;
32         totalNumOfBoxesSold = totalNumOfBoxesSold
33                               + numOfBoxesSold;
34         numOfStudents++;
35         cin >> name;
36     }
37     cout << endl;
```

```
39     cout << "Jumlah kotak yang terjual: "
40           << totalNumOfBoxesSold << endl;
41     cout << "Harga per kotak adalah Rp. ";
42     cin >> costOfOneBox;
43     cout << endl;
44     cout << "Total uang yang diperoleh dari hasil penjualan "
45           << "cookies adalah Rp. "
46           << totalNumOfBoxesSold * costOfOneBox << endl;
47
48     if (numOfStudents != 0)
49         cout << "Rata-rata jumlah kotak yang berhasil"
50               << " dijual oleh masing-masing mahasiswa: "
51               << totalNumOfBoxesSold / numOfStudents
52               << endl;
53     else
54         cout << "No input." << endl;
55     return 0;
56 }
```

## 3.5 Sentinel-Controlled **while** Loops (Lanj...)

Output:

```
Tuliskan nama penjual dan jumlah kotak yang terjual. Akhiri dengan -1 :  
Roxy 20  
Molly 15  
Melos 25  
-1  
  
Jumlah kotak yang terjual: 60  
Harga per kotak adalah Rp. 10000  
  
Total uang yang diperoleh dari hasil penjualan cookies adalah Rp. 600000.00  
Rata-rata jumlah kotak yang berhasil dijual oleh masing-masing mahasiswa: 20  
Program ended with exit code: 0
```

## 3.6 Flag-Controlled **while** Loops

- Cara ini menggunakan variable **bool** untuk menentukan kondisi benar atau salah, contohnya **isFound**, **isFull**, **isTallEnough**.
- Struktur:

```
isFound=false  
while (!isFound)  
{  
    ...  
    if (expression)  
        isFound = true;  
    ...  
}
```

## 3.6 Flag-Controlled while Loops (Lanj..)

```
9  #include <iostream>
10 #include <cstdlib>
11 #include <ctime>
12
13 using namespace std;
14 int main()
15 {
16     //declare the variables
17     int num;           //variable to store the random
18                       //number
19     int guess;         //variable to store the number
20                       //guessed by the user
21     bool isGuessed;    //boolean variable to control the loop
22     srand(time(0));
23     num = rand() % 100;
24     isGuessed = false;
25     while (!isGuessed)
26     {
27         cout << "Masukkan nilai integer lebih besar atau "
28              << "sama dengan 0 dan kurang dari 100: ";
29         cin >> guess;
30         cout << endl;
31         if (guess == num) {
32             cout << "Tebakan Anda benar. " << endl;
33             isGuessed = true;
34         }
35         else if (guess < num)
36             cout << "Tebakan Anda lebih kecil. \n Tebak lagi!" << endl;
37         else
38             cout << "Tebakan Anda lebih besar.\n Tebak lagi!" << endl;
39     } //end while
40     return 0;
41
42 }
```

## 3.6 Flag-Controlled while Loops (Lanj..)

Output:

```
Masukkan nilai integer lebih besar atau sama dengan 0 dan kurang dari 100: 50
Tebakan Anda lebih kecil.
Tebak lagi!
Masukkan nilai integer lebih besar atau sama dengan 0 dan kurang dari 100: 75
Tebakan Anda lebih kecil.
Tebak lagi!
Masukkan nilai integer lebih besar atau sama dengan 0 dan kurang dari 100: 82
Tebakan Anda lebih kecil.
Tebak lagi!
Masukkan nilai integer lebih besar atau sama dengan 0 dan kurang dari 100: 90
Tebakan Anda lebih besar.
Tebak lagi!
Masukkan nilai integer lebih besar atau sama dengan 0 dan kurang dari 100: 85
Tebakan Anda lebih kecil.
Tebak lagi!
Masukkan nilai integer lebih besar atau sama dengan 0 dan kurang dari 100: 87
Tebakan Anda lebih besar.
Tebak lagi!
Masukkan nilai integer lebih besar atau sama dengan 0 dan kurang dari 100: 86
Tebakan Anda benar.
Program ended with exit code: 0
```

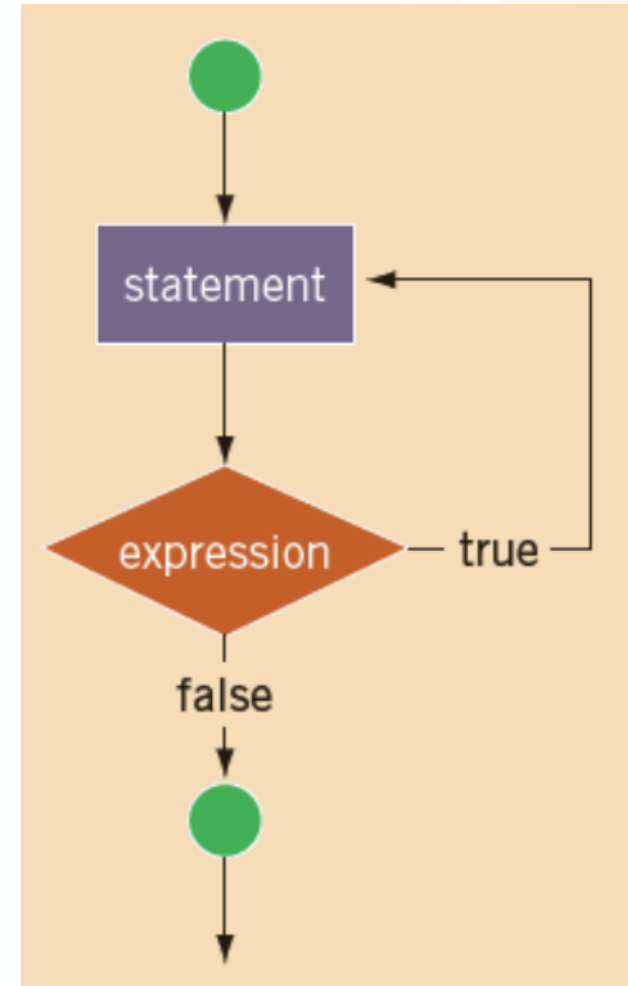


## **4. Konstruksi Repeat dalam bahasa C++**



## 4.1 Struktur

```
do  
    statement  
while (expression) ;
```



## 4.2 Contoh

```
9  #include <iostream>
10 using namespace std;
11
12 int main(){
13     int i;
14     i=0;
15     do
16     {
17         cout << i << " ";
18         i = i+5;
19     }
20     while(i<=20);
21     return 0;
22 }
```

Perhatikan bahwa setelah menampilkan output 20, maka instruksi  $i = i + 5$  dieksekusi, dan hasilnya adalah 25.

Oleh karena tidak memenuhi syarat  $i \leq 20$ , maka perulangan Berhenti.



0 5 10 15 20 Program ended with exit code: 0

## 4.3 Perbedaan do...while dan while...do

```
23 #include <iostream>
24 using namespace std;
25
26 int main(){
27     int i;
28     i = 11;
29     do {
30         cout << i << " ";
31         i=i+5;
32     }
33     while(i<=10);
34     cout << endl;
35     return 0;
36 }
```



11

Program ended with exit code: 0

```
9 #include <iostream>
10 using namespace std;
11
12 int main(){
13     int i;
14     i = 11;
15     while(i<=10) {
16         cout << i << " ";
17         i=i+5;
18     }
19     cout << endl;
20     return 0;
21 }
```



Program ended with exit code: 0

# Ringkasan

- Pengulangan dapat dilakukan dengan 3 cara yaitu: for, while-do, dan do-while (atau repeat-until).
- FOR merupakan jenis pengulangan tanpa kondisi (*unconditional looping*), sedangkan WHILE dan REPEAT akan mengeksekusi semua instruksi di dalam badan pengulangan sampai kondisi berhenti terpenuhi (conditional looping)

# LATIHAN/REVIEW MANDIRI

1. Apakah output dari program di bawah ini?

```
int i = 1;
int num = 2;
while (i < 5)
{
    num = num * i;
    i = i + 1;
}
cout << "i = " << i << " and num = " << num << endl;
```

# LATIHAN/REVIEW MANDIRI (Lanj..)

2. Apakah output dari program di bawah ini?

```
int count = 0;
int sum = 0;
while (count <= 5)
{
    sum = sum + count * (count - 1);
    count++;
}
cout << sum << endl;
```

# LATIHAN/REVIEW MANDIRI (Lanj..)

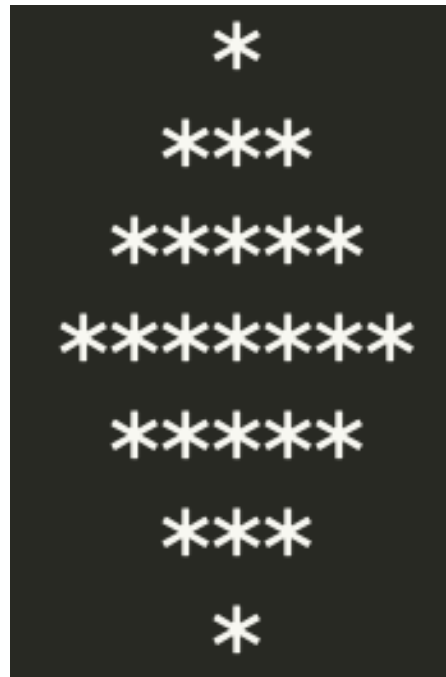
3. Apakah output dari program di bawah ini?

a. `for (i = 1; i <= 1; i++)  
 cout << "*";  
cout << endl;`

b. `for (i = 2; i >= 1; i++)  
 cout << "*";  
cout << endl;`

# LATIHAN/REVIEW MANDIRI (Lanj..)

4. Tulislah program untuk menampilkan bentuk seperti contoh di bawah ini:





# LATIHAN/REVIEW MANDIRI (Lanj..)

5. Tulislah program untuk menampilkan segitiga berupa angka seperti contoh di bawah ini:

```
1
232
34543
4567654
567898765
67890109876
7890123210987
890123454321098
90123456765432109
0123456789876543210
123456789010987654321
```



# Terima Kasih

U N I V E R S I T A S   B U N D A   M U L I A